Congested Scene Classification via Efficient Unsupervised Feature Learning and Density Estimation

Yuan Yuan, Jia Wan, Qi Wang*

School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.

Abstract

We present an unsupervised learning algorithm with density information considered for congested scene classification. Though many works have been proposed to address the general scene classification during the past years, congested scene classification is not adequately exploited yet. In this paper, we propose an efficient unsupervised feature learning approach with density information encoded to solve this problem. Based on spherical k-means, a feature selection process is proposed to eliminate the learned noisy features. Then, the local density information which better reflects the crowdedness of a scene is encoded by a novel feature pooling strategy. We evaluate the proposed method on the assembled congested scene data set and UIUC-sports data set, and intensive comparative experiments justify the effectiveness of the proposed approach.

Keywords: Computer vision, unsupervised feature learning, scene classification, density estimation, spherical k-means, feature pooling

1. Introduction

Public traffic has become a serious problem for the development of modern cities. In order to make the transport well organized, the primary task is to know the traffic status precisely. Among the various techniques enabling this function, congested scene classification is a critical one. But unfortunately, the ambiguity, variability and scale

*Corresponding author.
Email address: crabwq@nwpu.edu.cn (Qi Wang)

diversity in scenes make it a challenging task. In order to recognize different scenes, various methods have been proposed over the years. Among them, most literatures concentrate on the critical step of feature representation, which can be roughly divided into two categories: hand-craft features and learned features. Hand-craft features are widely used in scene classification since it is effective and interpretable [1, 2, 3]. Learned features (i.e. a transformation from raw image patches to more efficient representations) are generated by feature learning (FL) methods [4, 5, 6], which are thought to be adaptive to various situations. Because of the usefulness of FL, scene classification has been successfully used in applications such as aerial scene categorization [7], content-based image retrieval [8] and object detection [9], and has achieved notable performance in all these fields.

Congested scene classification is a specific problem of scene classification. Compared to traditional scene classification, we concern more about the crowdedness of pedestrians and vehicles in the traffic environment, with the purpose of labeling the scene image with a crowdedness category level. The increasing attention to this problem derives from the fact of congested traffic status. If we can get an understanding of the traffic scene by automatic analysis, the traffic management will be possible and easier [10]. Therefore, the monitoring of the transport status becomes essential. Under this circumstance, how to classify congested scenes correctly and effectively becomes a critical task.

Unfortunately, traffic congested scene classification is not adequately exploited yet. Only a few works concentrate on crowd density measurement [11, 12], traffic congestion classification [13] or crowd analysis [14]. In these tasks, the background subtraction and density estimation are the most important components. For background subtraction, optical flow is the most direct and effective method to be applied with static backgrounds. For density measurement, the local key points or individual detection are aggregated to estimate the number of objects. Nevertheless, the achieved performance is very limited.

Although many algorithms have been proposed to improve scene classification accuracy, they still have limitations when applied to congested scene classification task. On one hand, conventional approaches still have deficits. Hand-craft feature based

Figure 1: Typical images in the congested scene data set. From left to right, the three columns respectively represent the crowded scene, normal scene and open scene.

approaches are convenient, but they are not able to generalize to new environment. Unsupervised Feature Learning (UFL) based approaches achieve better performance, but most UFL algorithms in scene classification have many parameters to tune [15]. Moreover, the training stage is time-consuming which makes it inefficient in practical usage. On the other hand, prior information towards this particular application is not well considered. Since we care about the crowdedness of pedestrians and vehicles in a scene, the representation should encode density information, which is an efficient indication of congested scenes. Unfortunately, conventional approaches do not pay attention to this important information. For example, [16] utilizes density information by simply calculating the ratio between pedestrian and road area. At the same time, most approaches separately take pedestrians and vehicles into consideration. But in the real world, pedestrians and vehicles always appear simultaneously which makes the classification results less accurate.

To address congested scene classification and alleviate problems mentioned above, we first assemble a new data set which contains three different levels of congested scenes. Subsequently, an efficient unsupervised feature learning method is exploited for low-level features extraction. Based on the obtained feature prototypes, we further encode the density information into image representation to help discover the congested appearances in scenes. The major contributions of this paper are summarized as

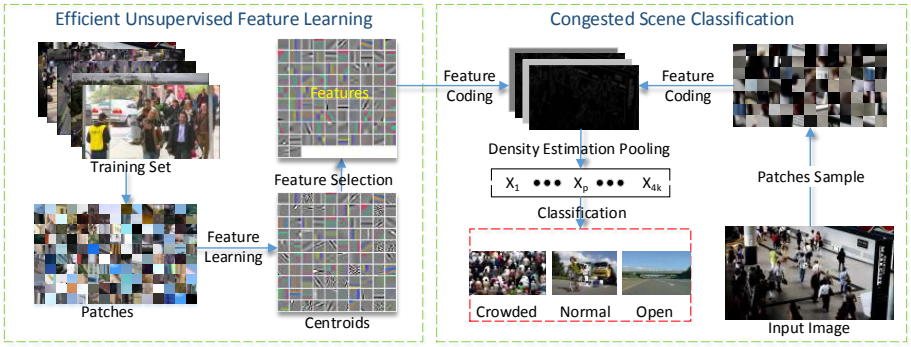


Figure 2: The pipeline of the proposed method. (1) **Feature learning.** Raw image patches are firstly extracted and a pre-processing is applied to remove the correlations between nearby pixels. After that, spherical k-means is employed to learn k centroids (i.e. features) and a feature selection procedure is followed to eliminate features which will misguide the classification result. (2) **Congested scene classification.** We first extract patches from an input image and LLC coding is used to map raw image patches to feature space (i.e. feature coding) and form k feature channels. Then, the ROI is detected since we only concern about pedestrians and vehicles in congested scenes. Subsequently, the local features are aggregated together by density estimation. Finally, a SVM classifier is trained for classification decision.

follows:

1. Since there is no available data set for congested scene classification, we assemble a new data set including three types of traffic scenes: crowded scene, normal scene and open scene. This data set will help exploit congested scene classification with dramatically changed backgrounds and provides a platform for the research community. Typical examples in this data set are shown in Fig. 1.

2. A more efficient approach based on spherical k-means and feature selection is proposed for congested scene classification. Recently, the most widely used UFL algorithm in scene classification is sparse auto-encoder (SAE) [7, 17, 8, 5]. However, the training of SAE is time-consuming, which makes it inefficient in practical usage. Motivated by this point, this paper exploits a more efficient UFL algorithm—spherical k-means. It does not have any parameters and the training is faster than other UFL algorithms. Besides, instead of taking all the learned features completely like traditional treatment, we refine them to eliminate the noisy ones with a feature selection post-processing. This strategy can ensure a high-quality feature utilization.

3. Different from conventional scene classification approaches, a density estimation method is proposed to encode density information. We do not need to detect all the pedestrians and vehicles. In contrast, we focus on directly modelling scenes by exploiting the variations in the local spatial arrangements of structural patterns captured by the learning procedure. Then a pooling operation is conducted to estimate the density according to the feature response. Therefore, our processing is more robust and reliable.

The remainder of this paper is organized as follows. Related work is elaborated in Section 2. The details of the proposed approach is followed in Section 3. The performance of the proposed method is reported in Section 4. Finally, conclusion and future work are presented in Section 5.

2. Related Work

We briefly review the relevant works which have made great progress for decades in scene classification. As aforementioned, literatures mainly concentrate on feature representation. As for the classification methods, Support Vector Machine (SVM) is the most frequently utilized classifier. Therefore, we will focus on the different features in the following.

Low-level features, such as histogram of color/texture and power spectrum [18], are initially explored in the early days in scene classification [19, 20]. It is simple and effective for binary classification (e.g., indoor versus outdoor, man-made versus natural). However, the so called "semantic gap" between low-level features and high-level semantic labels becomes the bottleneck for further improvement. Moreover, the accuracy drops significantly when category's size becomes large.

To overcome the limitation of low-level features, bag-of-feature (BOF) based models are proposed [21, 22, 23]. Basic BOF model [24] consists of feature learning and feature encoding. For the first step, the codebook is generated by clustering algorithm, typically k-means. After the codebook is generated, the input image is represented by histogram of unordered codewords. This model can encode the prior information and reduce the "semantic gap", which is significantly superior to low-level features. But

the spatial clue is lost as the histogram is orderless. Besides, with hard-assignment utilized, the information will be lost as well.

The feature encoding strategy in basic BOF model is inefficient. Thus more sophisticated feature encoding methods are proposed to reduce the information lost [25, 26]. These methods can be broadly divided into two steps: feature coding and feature pooling. In the coding step, instead of hard-assignment, sparse coding (SC) is included to achieve lower reconstruction error and make the representation specialized [27]. Locality-constraint coding (LLC) which is inspired by [28] is another coding method [29]. It is more smooth than SC because the similar local descriptors have similar codes by sharing bases. More extensive studies of coding methods can be found in [30]. In the pooling step, motivated by [31], Lazebnik *et al.* [32] proposed a spatial pooling method, namely Spatial Pyramid Matching (SPM), to encode spatial information efficiently. SPM divides the input image into spatial bins at different scales and then all histograms of visual-word in these bins are concatenated to produce the final representation. A more flexible pooling method is proposed in [33] recently which jointly learns appearance and important spatial pooling region (ISPR). This method improves the performance by reducing the influence of false responses as the representative objects of the scene appear at several important regions with high possibility. More variants to encode spatial information in BOF can be found in [34].

Compared to BOF based model, object filter based models utilize higher-level semantic features [2, 35, 36]. Intuitively, different scenes contain different typical objects. Thus, we can use the objects and the quantities of them to characterize different scenes efficiently. Motivated by this, many off-the-shelf object filters are trained to detect objects in different scenes. The responses of these filters are used for representation of the image. Usually, a spatial pyramid is employed to encode spatial information. These approaches often need long time to train, and the selection of typical objects is subjective.

Different from artificially designed features, feature learning is trying to automatically learn effective features from training data. Typical learning algorithms are k-means [15], sparse Restricted Boltzmann Machines (RBMs) [37], sparse auto-encoder [17, 5], and multi-layer based deep learning [38]. Although feature learning is powerful

to learn self-adaptive features and offers high classification performance, they usually have many hyper parameters to tune without a criterion to follow, which makes it inconvenient to use. Besides, the nature of the representation formed by deep networks and why it works so well are still to be explored.

3. Our Method

To remedy the problem associated with congested scene classification, this paper utilizes an efficient unsupervised feature learning method with density information encoded. As shown in Fig. 2, the proposed approach can be divided into two parts: efficient unsupervised feature learning and congested scene classification. In efficient unsupervised feature learning, raw image patches are first extracted. Then, we apply pre-processing on them to remove the correlations between nearby pixels. After that, spherical k-means is employed to learn features and a feature selection procedure is followed to remove bad features. In congested scene classification, Patches are first extracted from images. Then LLC is used to map raw image patches to a new feature space based on the previously learned feature codebook. With these, the local features are aggregated together by density estimation. Finally, a SVM classifier is trained for the classification decision.

3.1. Efficient Unsupervised Feature Learning

An unsupervised feature learning method and a feature selection procedure are used to learn and select useful features. We utilize spherical k-means as unsupervised feature learning as it is efficient to train and tune parameters. After feature learning, a simple and effective feature selection procedure is applied to reduce noises.

3.1.1. Data Augmentation

Due to the limitations of sample diversities and the availability of abundant training samples, it's necessary to augment the training data. The artificial enlargement of the training set is an easy and commonly used method to preclude over-fitting [39, 40]. We first transformed the images in training set to different scales and rotations. As a

result, an augmented training set is generated. We then random sample patches from the augmented data set.

Specifically, the data augmentation consists of two distinct forms, scale and rotation transformations. Since the pedestrians and vehicles in congested scenes are relatively small, we first enlarge each training image by 1.5 and 2 times. Then each obtained image is rotated by -15 and 15 degrees. With this processing, there are a total of 9 times amount of training images compared to the original training set. We then randomly sample raw patches in the training images. This makes the learned features more diverse and comprehensive.

3.1.2. Feature Learning

Generally, a pre-processing on samples before feature learning is necessary. We therefore normalize patches and then whiten them to remove the correlations between nearby pixels [41]. Given n original samples $X^o = \{x_1^o, x_2^o, ..., x_n^o\}$, they are first normalized by:

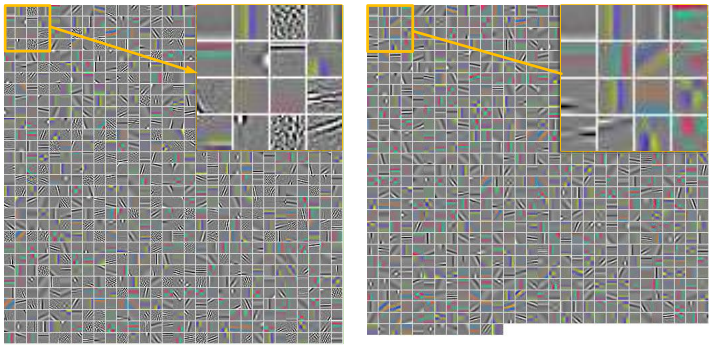
$$x_i^n = \frac{x_i^o - mean(x_i^o)}{\sqrt{var(x_i^o) + \epsilon_n}}, \forall i, \tag{1}$$

where ϵ_n is a small number to avoid being divided by zero and then are whitened by:

$$x_i^w = V(D + \epsilon_w I)V^\top x_i^n, \forall i, \tag{2}$$

where $[V, D] = eig(cov(X^n))$ and ϵ_w is a small constant. $X^n = \{x_1^n, x_2^n, ..., x_n^n\}$ are the normalized samples and $X^w = \{x_1^w, x_2^w, ..., x_n^w\}$ are the whitened samples. With this processing, the input patches are normalized and decorrelated.

Feature Learning is introduced to learn a transformation from raw image patches to representations that can be exploited more efficiently. We believe traditional feature descriptors destroy the structure of data somehow, while the learned features, especially deep features, can obtain more intrinsic and discriminative characteristics. However, the foundation of deep learning is the large training set. Since obtaining a large amount of data in real situation is not an easy task, it is impractical for deep network to learn useful parameters on existing data set. In this paper, a modified version of k-means which is called spherical k-means is utilized for unsupervised feature learning. It is a



(a) The learned feature centroids with noises.

(b) The remaining features after feature selection.

Figure 3: Comparison before and after feature selection. (a) Before feature selection, many noisy features exist. For example, the third features of the first line in the magnified block contains meaningless dotted structures. (b) After feature selection, the noisy features are removed. All remained features are edge detectors which can be clearly seen in the magnified block.

simple and efficient single-layered method in which less parameters are needed to tune and the training is faster than deep learning.

k-means clustering is widely used in computer vision aiming to partition n observations into k clusters. Given n samples $X = \{x_1, x_2, ..., x_n\}$ and k clusters $C = \{c_1, c_2, ..., c_k\}$. It is solved by minimizing the within-clustering sum of squares. The objective function is:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x \in c_i} |x - c_i|^2 \tag{3}$$

where c_i denotes the ith cluster.

Spherical k-means is different from k-means as the cosine similarity is used instead of Euclidean distance to measure the similarity from observations to centroid. This is because cosine similarity has been shown to be superior to Euclidean distance for high-dimensional data [42]. Specifically, given a patch $x_i \in \Re^N$ sampled from the training set, we want to learn a set of centroids $C \in \Re^{N \times k}$ which are used to map x_i to $s_i \in \Re^k$, where s_i denotes the learned feature of x_i. C is termed as *centroids* in this paper since the algorithm is derived from a clustering method.

After pre-processing, the spherical k-means can be formally expressed as:

$$\begin{aligned} &\min_{C,s} \sum_i |Cs_i - x_i|_2^2 \\ &\text{subject to } |s_i|_0 \leq 1, \forall i \\ &\text{and } |C_j|_2 = 1, \forall j, \end{aligned} \tag{4}$$

where $|s_i|_0 \leq 1$ denotes each x_i is assigned to only one of the centroids and $|C_j|_2 = 1$ means each codeword is unit length to prevent C_j from becoming too large or small. Given an initial C, this object function is solved by the following iterative steps:

1. Find maximum cosine similarity of each x_i and calculate the mapped code s_i by:

$$s_i = \begin{cases} C_j^\top x_i, & \text{if } j == \arg\max_l |C_l^\top x_i| \\ 0, & \text{otherwise,} \end{cases} \forall j, i, \tag{5}$$

2. Update C by:

$$C = XS^\top + C. \tag{6}$$

3. Normalize C_j by:

$$C_j = \frac{C_j}{|C_j|_2} \forall j, \tag{7}$$

until convergence. More details can be found in [41].

3.1.3. Feature Selection

Feature selection is utilized to choose useful features after feature learning. Usually, given a set of features, not all of them are useful. Noises often appear especially when the training set is small or parameters are not fine-tuned as shown in Fig. 3(a). In this case, some features may be redundant, and others may misguide the classification results. We thus propose an efficient and simple way to select the most effective features.

Different from traditional feature selection [43], we utilize a binary classifier to fulfill this task. As shown in Fig. 3, the appearances of useful features and noisy features are different. Motivated by this point, the proposed method can separate them effectively. In particular, we treat each centroid as a picture and an off-the-shelf classifier

is trained to classify the learned centroids which take the LBP [44] as the appearance feature. Negative centroids are generated when the training set is small while positive centroids are generated when the training images are enough and the parameters are fine-tuned. A SVM classifier is used as classifier. Noises are significantly reduced after feature selection as demonstrated in Fig. 3(b).

The pipeline of the full efficient unsupervised feature learning method is summarized in Algorithm 1.

Algorithm 1 Unsupervised feature learning

Input: The set of training images.

1: Training patches generation.
   (1) Enlarge the training images by 1.5 and 2 times scaling.
   (2) Rotate the training images by a -15 and 15 degrees.
   (3) Randomly sampling local patches from training images.
   (4) Normalize and whiten the obtained patches.
2: Apply Spherical k-means to learn feature mapping by Eq. 4. It can be solved by iterative Eq. 5-Eq. 7.
3: Reduce the learned centroid noise by feature selection.
   (1) Generate the negative and positive centroids under different training conditions.
   (2) Extract the LBP features of the centroids.
   (3) Train a SVM classifier using the obtained LBP features.
   (4) Select the positively classified centroids as the desired ones.

Output: Centroids matrix C mapping the raw image patches to a new feature space.

Although the proposed work utilizes a SVM classifier to remove noisy features, that doesn't make it a semi-supervised or supervised method. This is because the positive and negative samples for the SVM classifier are self-generated according to different settings described in Section 3.1.3, instead of employing the manually labeled ground truth. Therefore, we never use any labeled data for training in the proposed learning process, including feature learning and selection. As a result, we believe the proposed method is unsupervised considering the ways of utilizing data.

3.2. Congested Scene Classification

Now given an image as input, we first densely sample local patches from it. Subsequently, these patches are transferred to feature space (i.e., feature coding). Then the local features are dynamically pooled together steered by the region of interest (ROI) and the density is accordingly estimated. In the end, a linear SVM is adopted for the final decision.

3.2.1. Feature Coding

Feature coding aims to transfer local patches to learned feature space and generates more effective representations. LLC coding is utilized here as the representations generated by it are sparse and smooth, which are two preferable characteristics. The sparsity is ensured by assigning the small projection coefficients to 0 and the smoothness is constrained by an regularization term which assumes similar patches share similar bases. With the regularization, the correlations between local patches are captured. Formally, supposing we have n patches and k centroids, the LLC coding can be expressed as:

$$\begin{aligned} \min_{S} \sum_{i=1}^{n} |Cx_i - s_i|^2 + \lambda |d_i \odot s_i|^2 \\ \text{subject to } 1^\top s_i = 1, \forall i, \end{aligned} \tag{8}$$

where $S = \{s_1, s_2, ..., s_n\}$ refers to the codes of the local patches, $\odot$ denotes the element-wise multiplication,

$$d_i = \exp\left(\frac{dist(x_i, C)}{\sigma}\right), \tag{9}$$

where $dist(x_i, C) = [dist(x_i, c_1), dist(x_i, c_2), ..., dist(x_i, c_k)]$, and $dist(x_i, c_j)$ is the Euclidean distance between x_i and c_i. Note that d_i is usually normalized to $(0, 1]$. In practice, we utilize the approximate LLC to speed up the coding process. We first select several nearest neighbors of x_i and construct a local centroids $\tilde{C}$. Then, a linear

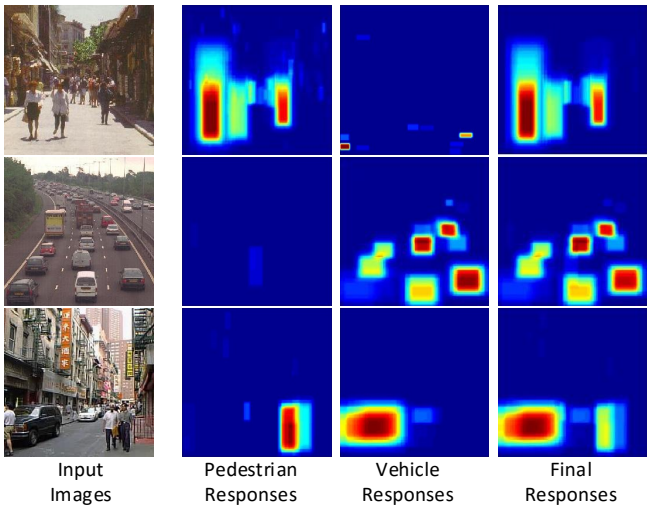


Figure 4: The typical response maps of congested scenes. The response map of pedestrians and vehicles are generated respectively and aggregated together to form the final response map.

function is solved to get the code. It can be expressed as:

$$\min_{S} \sum_{i=1}^{n} |\tilde{C}x_i - s_i|^2 \quad \text{subject to } 1^\top s_i = 1, \forall i. \tag{10}$$

Since $\tilde{C}$ is usually smaller than C, the computation is very efficient. LLC tends to encode a local patch with the centroids that are similar to it. To speed up the encoding process, an approximation is presented in [29]. Specifically, $\tilde{k}$ nearest centroids of a local patch are first chosen and form a local coordinate system. Then, Eq. 8 can be approximated through Eq. 10. The computation cost thus reduces from $O(k^2)$ to $O(k + \tilde{k}^2)$ where $\tilde{k} \ll k$. Through the approximation, the encoding process becomes more efficient and the locality in features is well preserved. More details can be found in [29].

3.2.2. Density Estimation for Pooling

After the feature coding, a density estimation pooling is followed to pool local features together and to estimate the crowdedness of pedestrians and vehicles at the same

time. Since the local features generated by feature coding are high-dimensional, we pool these features together to reduce the over-fitting problem and computational cost. After transferring raw patches to learned feature space, k (remember k is the centroid numbers) feature channels are constructed. Each channel can be seen as an individual response map of the input image. Since the density clue should be considered, we estimate the density of these feature channels through the maximum local sum response. The proposed method consists of ROI detection and estimation pooling.

Different from conventional approaches for density estimation, the ROI is detected at first as we only concern about pedestrians and vehicles in scenes. As a results, the effect of background can be eliminated and this makes the results more robust. Specifically, two filters based on Deformable Part-based Model (DPM) [45] are trained beforehand, one for pedestrians and the other for vehicles. The results of two filters are summed together to form the final response map. The two filters are just rough estimations of the ROI because finer judgement will be made subsequently. Then, the response map, namely heat-map, is produced by sliding filters across the input image. Typical heat-maps are shown in Fig 4.

With the obtained heat-map, we encode the density clue to efficiently represent the congested scene. We suppose a scene is congested if some local patches are full of pedestrians and vehicles. Thus, we concern more about the local maximum density of the feature channels. Particularly, we first divide each channel of image representation by 4 quadrants. In each region R, the relative heat-map is h. The maximum local density $D(R)$ is estimated to pool local features together. Formally, $D(R)$ can be calculated as follows:

$$D(R) = \max_{W} \sum_{x,y \in W} h(x,y) \cdot F_c(x,y) \tag{11}$$

where W denotes a sliding window in the subregion R and F_c is the examined feature channel. This equation implies that the maximum response within a window W is treated as the local density estimation of R. Since we have $4k$ regions in total, the final representation is $4k$ dimension. The illustration of density estimation pooling is shown in Fig 5.

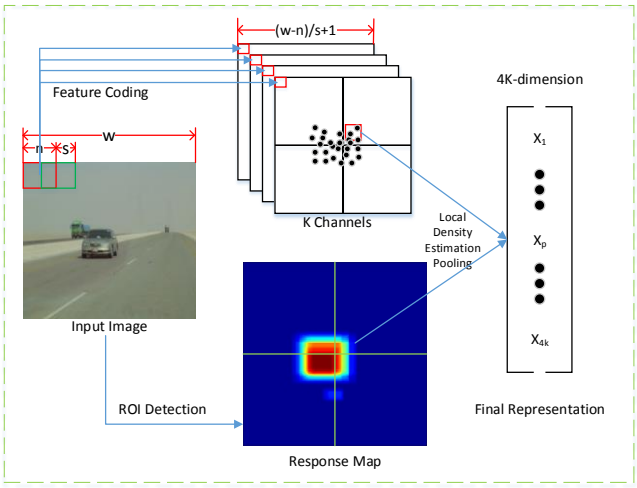


Figure 5: Illustration of density estimation pooling. After the feature channels and heat-map are generated, the local features are dynamically pooled together steered by ROI to form the final representation. The density is estimated at the same time.

A straightforward method for congested scene classification is by detecting the pedestrians and vehicles present in the scene. Note that the proposed method is different from that. The scheme based on object detection will make the framework highly depend on the accurate target detection preprocessing. At the same time, the computational complexity will increase dramatically. Motivated by this drawback, we choose an alternative that takes the image patches into consideration and predicts the density clue directly by analyzing the patches.

3.2.3. Classification

After we pool the local features together, the final representation of the input image is generated. Then, a SVM classifier is trained for final decision. Note that we adopt a linear kernel with the training costs on the order of $O(n)$ and the one-versus-rest strategy for multi-class classification.

The full routine of congested scene classification is summarized in Algorithm 2.

Algorithm 2 Congested scene classification

Input: The input image; the learned centroids matrix C.

1: Feature coding.

(1) Densely sample local patches from the input image.

(2) Transfer these local patches to feature space according to Eq. 8 and get k. feature response channels.

2: Density estimation pooling.

(1) Detect ROI roughly by off-the-shelf filters.

(2) Divide the input image into four quadrants and respectively pool the features in each subregion steered by ROI according to Eq. 11.

(3) Concatenate the four maximum responses for each channel to obtain a $4k$ dimension representation.

3: Classification.

(1) Train a linear SVM classifier.

(2) Classify the final representation to the semantic label.

Output: The semantic label.

4. Experiment

In this section, we first describe the data sets and parameters used for evaluating the proposed method. Then, the experimental results are introduced and discussed.

4.1. Data Sets

We evaluate the proposed approach on two data sets: congested scene data set and UIUC-Sports. The first data set is used to evaluate the performance of the proposed method for congested scene classification. The other data set is widely used in scene classification, and it is employed to evaluate the performance of the efficient unsupervised feature learning.

- Since there exist no data sets for congested scene classification, we assembled a new data set NWPU-CS from [46, 19] and the Internet. This data set consists of 1131 images divided into three categories. The first category is crowded scene which contains extremely congested pedestrians or vehicles. The second category is the open scene that has very few pedestrians or vehicles. The third one that has moderate amount of pedestrians and vehicles are called normal scenes.

The average image size is 600×400 and the typical examples in this data set are shown in Fig. 1.

We randomly select 80 images per class for training and the rest for testing in our experiments.

- The UIUC-sports [47] is a challenging data set containing 8 sporting scenes with 1579 RGB images. Most of the images are high-quality with the highest resolution up to several thousands pixels in one dimension. In addition, the backgrounds of images in this data set are much cluttered and the objects are diverse. Following the experimental settings in [47], for each class, 70 images are used for training and 60 images are used for testing.

4.2. Experimental Setup

The experiments are divided into two parts. The first part is used to evaluate the proposed system and the second part is used to evaluate the components of the proposed framework. During these experiments, there are several parameters to set. The patch size is set as 16×16 pixels. Thus, gray patches will form $16 \times 16 = 256$ dimensional vectors and RGB patches will form $16 \times 16 \times 3 = 718$ dimensional vectors. The patch sampling stride is set as 1 pixel which is proven to be efficient by Coates and Ng [41]. The window size w in density estimation is set as 4 by our experiment. This can be verified by experiments shown in Fig 6. To select the size of the learned feature centroids, we have done experiments to evaluate its influence as shown in Fig 7. We set the selected feature centroids to different sizes and then compare their classification performance. The results show that the best performance is obtained at the feature size close to 4000 for the feature learning process. Based on this, we set the learned feature size as 4000 in the following experiments.

4.3. Results

To evaluate the performance of the proposed method for congested scene classification, we apply it on NWPU-CS data set and compare the final performance with some conventional scene classification approaches—BOF, ScSPM and LLC. The BOF

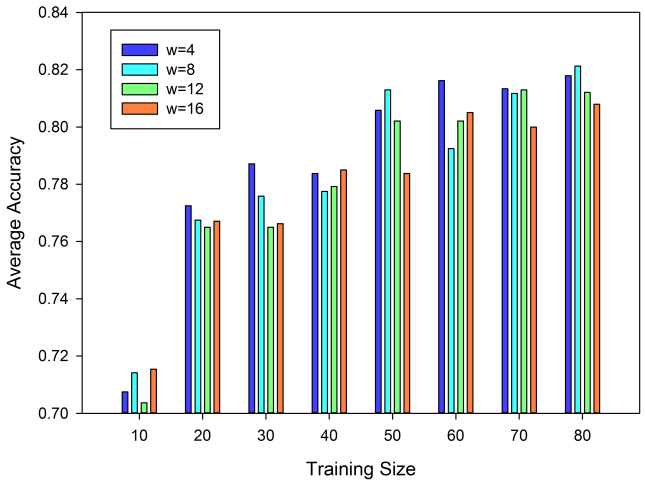


Figure 6: Classification results with different window sizes w. The horizontal axis is the training size. The vertical axis is the average classification accuracy.

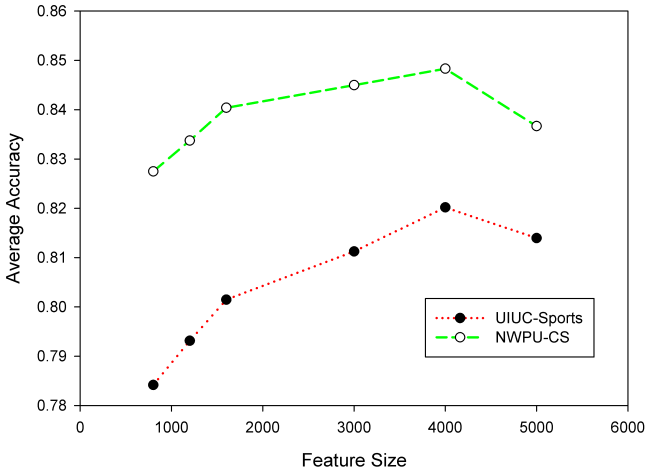


Figure 7: The effect of feature sizes on different data sets. The horizontal axis is the learned feature size. The vertical axis is the average accuracy for the classification results.

Table 1: Comparison of different methods on NWPU-CS data set.

Method	BOF	ScSPM	LLC	Ours
Accuracy	78.09%	80.58%	80.92%	85.50%

utilizes dense SIFT [48] as low-level features, hard-assignment for feature coding and Sum Pooling (SP) to aggregate local features together. Similarly, the ScSPM also utilizes dense SIFT as low-level features, but Sparse Coding (SC) and Spatial Pyramid Pooling (SPM) are used for feature encoding. As for LLC, it quite resembles to ScSPM with the only difference that the local constraint is additionally utilized for feature coding in ScSPM. In all experiments, the same codebook is utilized for the three algorithms. It is constructed by k-means and the size is set as 1024. Linear SVM is finally adopted as the discriminative classifier. Like most previous works, we repeat experiments 10 times and report the average accuracies. The final classification results are shown in Table 1. The proposed algorithm achieves 85.50% average accuracy and outperforms the traditional approaches.

We further show in Fig. 8 the performance of these methods under different training sizes. The results confirm that the proposed method yields better performance than conventional competitors. Compare to BOF, the proposed method not only learns more sophisticated features with lower-reconstruction error, but also characterizes the spatial arrangement of the features. Though ScSPM has better spatial information included, it does not involve a sophisticated encoding method. LLC tackles this problem well but it fails to consider the density information.

We further expand the proposed method to the general purpose scene classification, namely Efficient Unsupervised Feature Learning (EUFL). EUFL still employs the proposed feature learning and encoding step but with a more general sum pooling. Similarly, we compare EUFL with BOF, ScSPM and LLC but this experiment is performed on a widely used scene data set: UIUC-Sports. It is a challenging data set containing high-resolution RGB images. The overall accuracies are shown in Table 2 and Fig. 9. The results prove that EUFL is superior to the conventional approaches. For a detailed understanding of EUFL performance, the confusion matrix is shown in Fig.

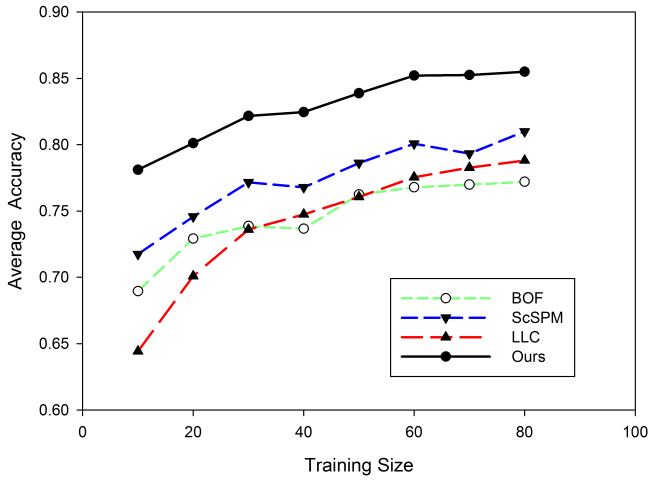


Figure 8: The classification results with different methods. The horizontal axis is the training size. The vertical axis is the average accuracy.

Table 2: Comparison of different methods on UIUC-Sports data set.

Method	BOF	ScSPM	LLC	EUFL
Accuracy	79.79%	83.44%	83.96%	86.44%

10. From the figure, it is manifest that most of the time, EUFL can demonstrate a good classification result. But for some classes, the accurate classification are difficult. The most confused classes are Croquet and Bocce which are very similar to each other. Another pair of confusion classes is Rowing and Sailing which have similar backgrounds. These cases are hard for not only the proposed method but also the other competitors.

These results confirm the fact that EUFL is efficient to learn useful features by minimizing the reconstruction error. Besides, after feature selection, the noise is significantly reduced. That makes the proposed approach effective and outperforms some hand-craft features.

4.4. Components Evaluation of The Proposed System

In this subsection, we investigate each component of the proposed system including feature learning, feature selection, feature coding and density estimation pooling.

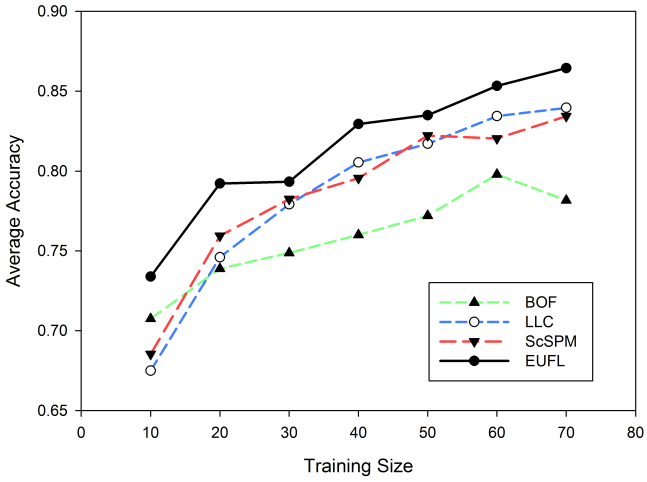


Figure 9: The effect of feature learning. Experimental results with different feature sizes on UIUC-Sports data set.

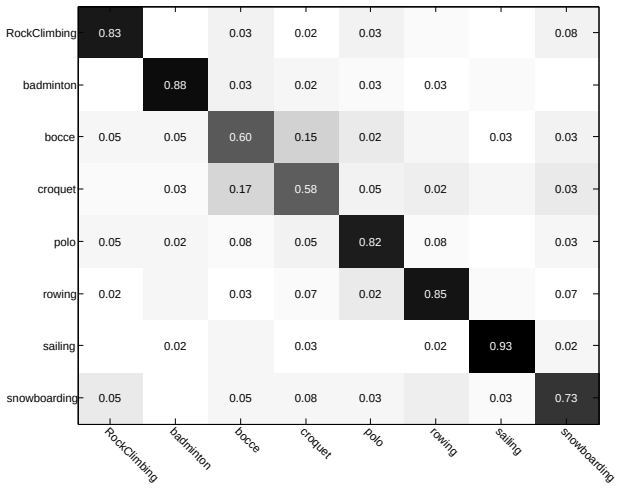


Figure 10: Confusion matrix on UIUC-sports data set by the EUFL method.

4.4.1. Evaluation of Feature Learning

Feature learning is the most important component for the proposed system. In this experiment, to exploit the efficiency of spherical k-means, we compare its training time with Independent Component Analysis (ICA) and Sparse Auto-encoder which are the most frequently taken methods in scene classification. The experiment is performed on 10000 RGB image patches of the size 32×32 ($32 \times 32 \times 3 = 3702$ dimensional inputs).

The training time is evaluated by the convergence of each algorithm, stopping when the change of the function value drops below a specific threshold. The results are shown in Fig. 11, from which the curves show the proposed feature learning algorithm is more efficient than the Sparse Auto-encoder and ICA. Note that Sparse Auto-encoder is not convergent in reasonable time (3 hours) for large feature sizes, which is intuitive that it takes significant long time especially with the feature size grows since the Sparse Auto-encoder requires more time to solve the l_1-regularized least squares problem. The ICA is faster than Sparse Auto-encoder but the orthogonalization in each iteration is time-consuming, which makes it slower than Spherical k-means. Since the number of independent components to be estimated range from $1-3701$, the maximum feature size of ICA in Fig. 11 is 3000 because the feature size of ICA must be less than the dimension of input. Recall that the spherical k-means can be solved by Eq. 5-Eq. 7, each step of which can be done quickly. Thus, the proposed algorithm is more efficient for feature learning.

4.4.2. Evaluation of Feature Selection

Feature selection is proposed to eliminate the influence of noises since not all of the learned features are useful. Considering the training images are often insufficient, we exploit how to efficiently generate robust features under this circumstance by the effort of feature selection. The experiment is first performed on NWPU-CS data set. The result can be seen in Fig. 12. We notice that when the feature size is small, the average accuracy will drop after feature selection. But with the increase of the feature size, the improvement through the feature selection appears. This is because when the feature size increases, it is more difficult to learn effective features especially when the

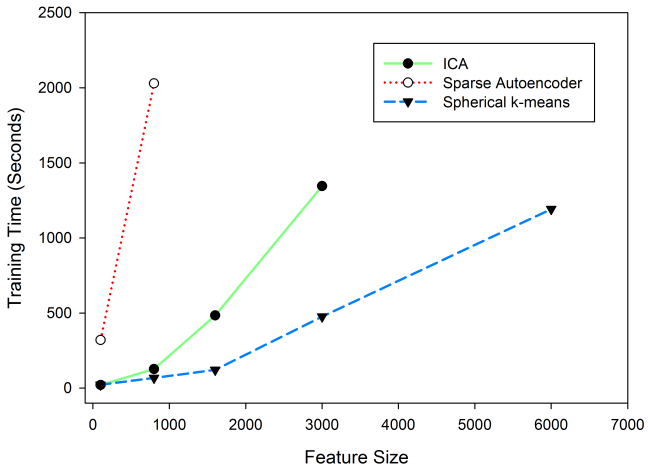


Figure 11: Comparison the proposed feature learning on NWPU-SC data set. The horizontal axis is the learned feature size. The vertical axis is the training time.

training images are not enough. Thus an effective feature selection is necessary. But the curve drops down when the feature size is more than 4000. This is unsurprising that at this time, the over-fitting occurs since the training set is limited compared to the feature size. Therefore, we set the feature size to 4000 in our experiments.

4.4.3. Evaluation of Feature Coding

Feature coding is an essential component of image representation, whose intention is to map raw image patches to learned feature space. It considerably affects the representation effectiveness and computation cost. In order to justify the effectiveness of the employed LLC coding, we compare it with hard-assignment and triangle coding, keeping the other system components unchanged in the NWPU-CS data set. The classification results are shown in Fig 13. From the curves, it is obvious that the best performance is obtained when LLC coding is utilized. Hard-assignment is the simplest coding method which assigns the patch to the nearest centroid. Obviously, this strategy is easy to compute but will lost profuse information. Triangle coding is much softer than hard-assignment and somewhat ensures the sparsity but the codes produced by it are not smooth. Fortunately, LLC coding is an efficient coding method which guarantees the sparsity and smoothness at the same time. Therefore, LLC coding achieves the

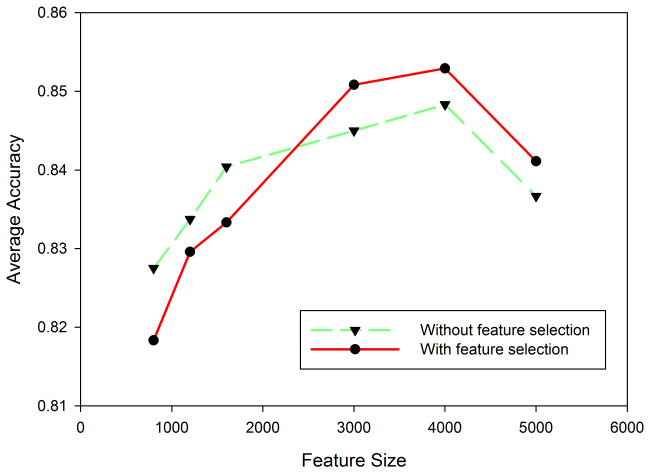


Figure 12: The effect of feature selection. The horizontal axis is the learned feature size. The vertical axis is the average accuracy for the classification results.

best performance than the other competitors.

4.4.4. Evaluation of Density Estimation Pooling

The density estimation pooling is proposed to encode density clue and make the final representation more efficient for the specific problem. To evaluate the effect of density estimation pooling, we compare it with sum pooling and max pooling. Similarly, we fix all the experimental setup except the pooling strategy and then compare their performance on the NWPU-CS data set. In the first place, an illustrative explanation of the density pooling is shown in Fig. 14 for a more straightforward understanding. From left to right, the three columns respectively represent the feature channels of different congested scenes, the results of density estimation pooling and sum pooling. It is easy to notice that the first scene is more congested than the second scene. But the sum pooling will be confused about their densities because it generates the same pooling results. Nevertheless, density estimation pooling can separate them effectively. From the figure we can see obviously that density pooling may generate different results from traditional pooling methods such as sum pooling. For example, the sum pooling cannot distinguish the situations in the first row and second row because it produces the same results. On the contrary, the proposed density pooling can effectively characterize their

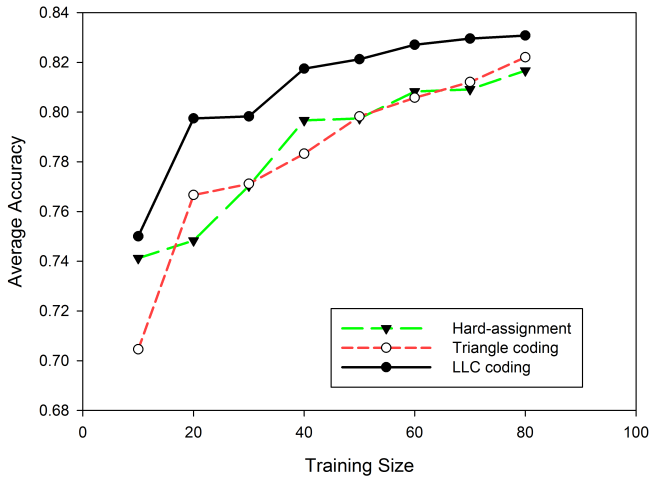


Figure 13: Comparison of different feature coding methods. The horizontal axis is the training size. The vertical axis is the average accuracy.

difference. For a more objective evaluation, Fig. 15 shows the average accuracies of different pooling methods under different training sizes. Through all the experiments, density estimation pooling outperforms other pooling methods for the congested scene classification task.

5. Conclusion

Congested scene classification is a specific problem of scene classification in which we concern more about the crowdedness of pedestrians and vehicles in the traffic environment. Though great progress has been made in general purpose tasks, no algorithms are designed towards this particular application. Based on the deficiency of existing research, we propose an efficient unsupervised feature learning and selection strategy to get the effective representation of the input image. The proposed feature selection method motivated by the different appearance of centroids can ensure a high-quality feature utilization. Furthermore, a more sophisticated method to pool features together based on local density estimation is presented. Our method shows its efficiency in congested scene classification task compared to some conventional scene classification algorithms.

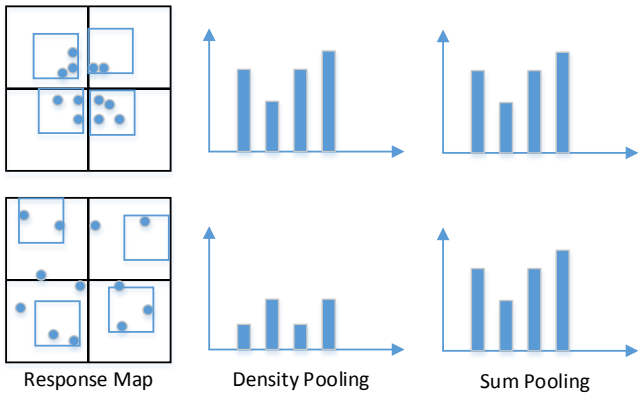


Figure 14: Illustration of the proposed density pooling and traditional sum pooling.

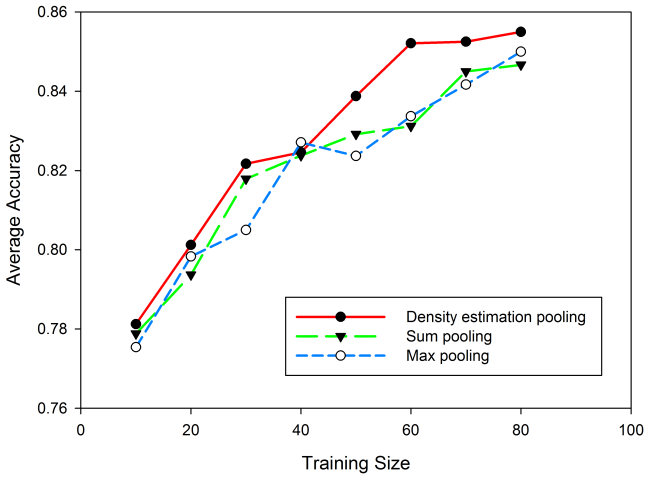


Figure 15: Comparison of density estimation pooling and conventional pooling methods. The horizontal axis is the training size. The vertical axis is the average accuracy.

Since the proposed approach is based on low-level features which are difficult to encode semantic information, a hierarchical model shall be exploited to learn high-level representations in the future. Besides, the temporal information shall be considered too.

References

[1] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognition 46 (2) (2013) 483–496.

[2] N. Serrano, A. E. Savakis, J. Luo, Improved scene classification using efficient low-level features and semantic cues, Pattern Recognition 37 (9) (2004) 1773–1784.

[3] A. Bolovinou, I. Pratikakis, S. Perantonis, Bag of spatio-visual words for context inference in scene classification, Pattern Recognition 46 (3) (2013) 1039–1053.

[4] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, Exemplar based Deep Discriminative and Shareable Feature Learning for scene image classification, Pattern Recognition 48 (10) (2015) 3004–3015.

[5] H. Yin, X. Jiao, Y. Chai, B. Fang, Scene classification based on single-layer sae and svm, Expert Systems with Applications 42 (7) (2015) 3368–3380.

[6] Z. Jiang, Z. Lin, H. Ling, F. Porikli, L. Shao, P. Turaga, Discriminative feature learning from big data for visual recognition, Pattern Recognition 48 (10) (2015) 2961 – 2963.

[7] A. M. Cheriyadat, Unsupervised feature learning for aerial scene classification, IEEE Transactions on Geoscience and Remote Sensing 52 (1) (2014) 439–451.

[8] J. A. Vanegas, J. Arevalo, F. A. Gonzalez, Unsupervised feature learning for content-based histopathology image retrieval, in: International Workshop on Content-Based Multimedia Indexing, 2014, pp. 1–6.

[9] J. Yao, S. Fidler, R. Urtasun, Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 702–709.

[10] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, L. D. Stefano, Traffic sign detection via interest region extraction, Pattern Recognition 48 (4) (2015) 1039 – 1049.

[11] S.-Y. Cho, T. Chow, C.-T. Leung, A neural-based crowd estimation by hybrid global learning algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29 (4) (1999) 535–541.

[12] G.-J. Kim, K.-Y. Eom, M.-H. Kim, J.-Y. Jung, T.-K. Ahn, Automated measurement of crowd density based on edge detection and optical flow, in: International Conference on Industrial Mechatronics and Automation, Vol. 2, 2010, pp. 553–556.

[13] A. Sobral, L. Oliveira, L. Schnitman, F. Souza, Highway traffic congestion classification using holistic properties, in: International Conference on Signal Processing, Pattern Recognition and Applications, 2013.

[14] Y. Yuan, J. Fang, Q. Wang, Online anomaly detection in crowd scenes via structure analysis, IEEE Transactions on Cybernetics 45 (3) (2015) 548–561.

[15] A. Coates, A. Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.

[16] S. Hua, J. Wua, L. Xub, Real-time traffic congestion detection based on video analysis, Journal of Information and Computer Sciences 9 (10) (2012) 2907–2914.

[17] F. Zhang, B. Du, L. Zhang, Saliency-guided unsupervised feature learning for scene classification, IEEE Transactions on Geoscience and Remote Sensing 53 (4) (2015) 2175–2184.

[18] A. Oliva, A. B. Torralba, A. Guérin-Dugué, J. Hérault, Global semantic classification of scenes using power spectrum templates, in: International Conference on Challenge of Image Retrieval, 1999, pp. 9–9.

[19] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.

[20] J. Wu, J. Rehg, Centrist: A visual descriptor for scene categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2011) 1489–1501.

[21] A. Bosch, A. Zisserman, X. Muoz, Scene classification using a hybrid generative/discriminative approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (4) (2008) 712–727.

[22] A. Singh, Parmanand, Saurabh, Survey on plsa based scene classification techniques, in: International Conference on Confluence The Next Generation Information Technology Summit, 2014, pp. 555–560.

[23] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566.

[24] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2005, pp. 524–531.

[25] S. Bai, X. Wang, C. Yao, X. Bai, Multiple stage residual model for accurate image classification, in: Asian Conference on Computer Vision, Vol. 9003 of Lecture Notes in Computer Science, 2015, pp. 430–445.

[26] X. Wang, X. Bai, W. Liu, L. Latecki, Feature context for image classification and object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 961–968.

[27] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[28] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Advances in Neural Information Processing Systems, 2009, pp. 2223–2231.

[29] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[30] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: a comprehensive study., IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 493.

[31] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, in: IEEE International Conference on Computer Vision, Vol. 2, 2005, pp. 1458–1465.

[32] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

[33] D. Lin, C. Lu, R. Liao, J. Jia, Learning important spatial pooling regions for scene classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3726–3733.

[34] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial-bag-of-features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3352–3359.

[35] Y. Zhao, Q. Wang, Y. Yuan, Action recognition based on semantic feature description and cross classification, in: IEEE China Summit International Conference on Signal and Information Processing, 2014, pp. 626–630.

[36] L.-J. Li, H. Su, L. Fei-Fei, E. P. Xing, Object bank: A high-level image representation for scene classification & semantic feature sparsification, in: Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.

[37] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted boltzmann machines for collaborative filtering, in: International Conference on Machine learning, 2007, pp. 791–798.

[38] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Advances in Neural Information Processing Systems, 2014, pp. 487–495.

[39] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642–3649.

[40] P. Simard, D. Steinkraus, J. C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: International Conference on Document Analysis and Recognition, 2003, pp. 958–963.

[41] A. Coates, A. Ng, Learning feature representations with k-means, in: Neural Networks: Tricks of the Trade, Vol. 7700 of Lecture Notes in Computer Science, 2012, pp. 561–580.

[42] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Workshop on Artificial Intelligence for Web Search, 2000, pp. 58–64.

[43] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Transactions on Cybernetics 44 (6) (2014) 793–804.

[44] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[45] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[46] J. Shao, C. Loy, X. Wang, Scene-independent group profiling in crowd, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2227–2234.

[47] L.-J. Li, L. Fei-Fei, What, where and who? classifying events by scene and object recognition, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[48] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

Author Biography

Yuan Yuan is currently a Full Professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xian, China. She has authored or coauthored over 150 papers, including about 100 in reputable journals such as IEEE Transactions and Pattern Recognition, as well as conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.

Jia Wan is currently working toward the M.E. degree in the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His current research interests include image classification and congestion analysis.

Qi Wang received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005 and 2010 respectively. He is currently an associate professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.