Neurocomputing 173 (2016) 355–363

Contents lists available at ScienceDirect

Neurocomputing

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

Describing and learning of related parts based on latent structural model in big data

Lei Liu [a], Xiao Bai [b,*], Huigang Zhang [b], Jun Zhou [c], Wenzhong Tang [b]

[a] College of Medical, Shantou University, Shantou 515063 and School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[b] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[c] School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia

ARTICLE INFO

Article history:
Received 26 June 2014
Received in revised form
25 November 2014
Accepted 14 December 2014
Available online 2 September 2015

Keywords:
Latent structural model
Big data
Incremental learning
Multiple kernel learning

ABSTRACT

In this paper, we propose a novel latent structural model for big data image recognition. It addresses the problem that large amount of labeled training samples are needed in traditional structural models. This method first builds an initial structural model by using only one labeled image. After pooling unlabeled samples into the initial model, an incremental learning process is used to find more candidate parts and to update the model. The appearance features of the parts are described by multiple kernel learning method that assembles more information of the parts, such as color, edge, and texture. Therefore, the proposed model considers not only independent components but also their inherent spatial and appearance relationships. Finally, the updated model is applied to recognition tasks. Experiments show that this method is effective in handling big data problems and has achieved better performance than several state-of-the-art methods.

1. Introduction

Due to the exploding data available on the Internet, tremendous interests in developing big data machine learning methods have emerged in recent years [1–3]. One of the applications in this area is annotation of large scale Web images, for which several datasets containing large number of images collected from the Internet have been created, for instance, Caltech datasets [4], Pascal VOC datasets [5] and ImageNet datasets [6]. Many computer vision and pattern recognition methods have been developed to analyze and annotate big image data. These include classification or ranking methods based on K-Nearest Neighbors (KNN) [7], Support Vector Machines (SVMs) [8], regression models [9], and deep learning [10]. Some methods have used hierarchy strategy to analyze big data problems. Hwang et al learned a tree of semantic kernels, where each node has a Mahalanobis kernel optimized to distinguish the classes in its children nodes [11]. Gao and Koller utilized a hierarchy structure in which image classes are divided into positive and negative groups separated by a binary classifier [12]. In [13,14], the authors also used matrix ideas to perform recognition tasks by dimensional reduction or matrix decomposition.

One of the challenges in dealing with large scale computer vision problems is how to develop effective and efficient feature representation. Many methods adopted the Bag-of-Words (BoW) method [15] which is a vector quantization-based approach. Locally normalized histogram of gradient (HOG) [16] is also a widely used method describe objects, especially for detection tasks [17]. In [18], a fully affine invariant Speeded Up Robust Features (SURF) was proposed to introduce the affine invariant property to SURF feature, while maintaining the feature's own advantage. Recently, Cheng et al. proposed a generic measure for objectness estimation, which is proved to be simple, fast, and effective [19]. Zhao et al. developed a feature coding method based on structural information of local descriptors [20]. In their work, 3D shape context has been incorporated into local spatio-temporal interest point features for human action recognition. Such hierarchical feature coding idea has also been shared by other researchers working on image classification [21].

Besides extracting local features for object description, many methods explore the importance of structural context information in an object, which lead to a batch of structural modeling methods in the literature [22–24]. Lee and Grauman proposed a graph based algorithm that models the interactions between familiar categories and unknown regions, which is used to discover novel categories in unlabeled images [25]. Yang et al. converted an image into a close-loop graph with super pixels as nodes [26]. Saliency can than be determined by ranking these nodes based on their similarities to background and foreground queries. In [27], a multi-

[*] Corresponding author.
E-mail address: baixiao@buaa.edu.cn (X. Bai).

http://dx.doi.org/10.1016/j.neucom.2014.12.120
0925-2312/© 2015 Elsevier B.V. All rights reserved.

feature fusion method was developed based on semantic similarity for image annotation.

Such structure based methods overcome the shortcomings of missing spatial information in the statistical methods. Part-based model is a special class of structural methods, where structure nodes represent visual parts and graph edges represent the spatial relations between these parts. In [28], Felzenszwalb et al. presented a deformable part model, which combines the part model with latent SVM method to get better recognition results. This method was extended by Ott and Everingham [29], which allows sharing of object part models among multiple mixture components as well as object classes.

Although many part-based methods use human to manually label training samples, in recent years, some efforts have turned to find semantic parts automatically. For example, Singh et al. used an iterative procedure that alternates between clustering and training classifiers, so as to discover a set of discriminative parts [30]. Endres proposed a method to learn a diverse collection of discriminative parts by relocating the object boxes while recognition [31]. Maji and Shakhnarovich presented a method for semi-supervised discovery of semantically meaningful parts from pairwise correspondence annotations [32].

A problem of the part-based methods is that their accuracy may not be guaranteed in case of insufficient labeled training samples. Therefore, some methods used hierarchical or incremental learning schemes to update and enrich the initial trained model [33]. In [34], Zhu and Shao introduced a weakly supervised cross-domain dictionary learning method that uses weakly labeled data from other visual domains as the auxiliary source data for enhancing the initial learning system. Zheng et al. proposed an online incremental learning SVM for large data sets, which consists of learning prototypes and learning support vectors [35]. Chen et al. presented an efficient alternative implementation of incremental learning [36]. It not only improves image processing performance, but also adapts to large datasets. In [37], incremental training of SVM was used as the underlying algorithm to improve the classification time efficiency. Pang et al proposed an incremental learning method that can not only incrementally model the features but also estimate the threshold and training error in a close form [38].

Above all, most of the work done in big data research area have focused on developing fast and efficient recognition algorithms [39]. In this paper, we show how to improve recognition accuracy on top of existing big data techniques. The aim of our work is to develop a latent structured part-based model which uses the inherent relationship between parts to describe objects. Furthermore, our method can extract candidate object parts given only one labeled training image, which is very suitable for big data problems with limited training samples. Different from most structural models, the main contribution of this paper is three-fold. Firstly, we propose a novel model formulation that mines the deep relationship (both appearance and spatial relationships) between parts in the object, while most previous works assume parts are independent of each other. Secondly, we present a part finding algorithm which learns a diverse collection of discriminative parts. It only needs one labeled training sample and can save human labelling costs in the training process. Thirdly, we introduce the multiple kernel learning method to describe parts. It enriches the distinctive part information, and therefore, makes the part matching results more accurate.

The rest of the paper is organized as follows. We first present the latent structural model formulation in Section 2. Then we describe the detailed feature extraction and representation method in Section 3. Next, the part finding and learning strategy is introduced in Section 4. Section 5 reports extensive experimental results that validate the effectiveness of the proposed model in big data recognition problems. Finally, we conclude the paper in Section 6 and propose our future work.

2. Latent structural model formulation

In order to predict objects in huge amount of images, we need to build models that can represent these objects. Here we propose a latent structural model that accounts for all the parts and their relationships in the object. This model is enlightened by the discriminative attribute model proposed by Wang and Mori [40] which is a multi-class object classifier that uses attributes as hidden variables. The relationships between the object categories and the attributes are described as learning parameters.

Let a training sample be represented as a tuple (x,h,y), where x is the training image, y is the label, and $h=(h1,h2,\ldots,hm)$ indicates m parts of an object in the image. The classifier $fw:\mathcal{X}\times\mathcal{Y}\rightarrow\mathbb{R}$ is parameterized by vector w. The proposed object model is defined as follows:

$$wT\Phi(x,h,y)=w^{yT}\phi(x;y)+\sum_{j}w^{hj}T\varphi(x;j,hj)+\sum_{(j,k)}w^{jkT}\psi(hj,hk)+\sum_{j}\tilde{w}^{hj}T\nu(x;j,hj) \quad (1)$$

In this equation, $w=\{wy,whj,wjk,\tilde{w}hj\}$ are the concatenation of the first parameter in each factor, and the other terms are the features composing $\Phi(x,h,y)$. These terms are defined as follows:

Object classifier $w^{yT}\phi(x;y)$: It is a standard linear model for object recognition without considering object parts. $\phi(x;y)$ is the probability that image x has label y, which can be obtained by training a multi-class SVM.

Part classifiers $w^{hj}T\varphi(x;j,hj)$: It is a standard part model trained to predict the label of part j for image x. It is an independent part for object prediction without considering the object itself or other parts. $\varphi(x;j,hj)$ is the probability that part j is labeled as h_j, achieved by training a binary SVM for this part.

Part/object – part interaction (appearance level) $w^{jkT}\psi(hj,hk)$: It gives the appearance relationship between the j-th part and the k-th part. Furthermore, if we define the 0-th part as the object itself, this model can also give the relationships between the whole object and its parts. $\psi(hj,hk)$ is the joint probability between two parts. It can be achieved by using the part probabilities according to their appearance descriptors.

Object – part interaction (spatial level) $\tilde{w}^{hj}T\nu(x;j,hj)$: It represents the spatial relationships between the j-th part and the object center. $\nu(x;j,hj)$ is a learned variable that gives the location information of part j. In the testing phase, it can be calculated by measuring the spatial overlap ratio between the testing part j and the model part h_j. Furthermore, this term can also describe the spatial relationships between two parts h_k and h_j. Thus, $\nu(x;j,hj)$ can be re-written as $\nu(hk,hj)$, which gives the relative locations between each two parts. However, for computational efficiency, we do not use the spatial interactions between parts in this paper.

Although the above model is based on the attribute model in [40], there are some fundamental differences. The goal of our model is describing the part–part relationships and the part–object relationship. First, we propose the part concept instead of attributes, which makes our method more like a latent structural model. Second, we consider the spatial information of parts, and use them to expand the model. This is not a component in [40]. Third, the model in [40] is further modified by combining the object-part and part-part interactions. Last and most important, we develop novel formulation to represent the last two terms in Eq. (1). This allows all four terms use probabilities as measurements. Furthermore, we also introduce an incremental learning process to update the model, which will be discussed later.

So far, we have got the structural model for object recognition. Assume w is known, the recognition problem is then turned into solving the following equation:

$$h* = \arg\max_{h} wT\Phi(x,h,y) \tag{2}$$

After getting the optimal $h*$ we can estimate the parts' locations, and then recover the location of object itself.

To learn the model in Eq. (2), we follow the approach proposed in [40]. Given a set of training samples $\{(x,h,y)\}$, h is treated as a latent variable. Therefore, the training problem can be transferred to a latent SVM formulation as follows:

$$\begin{aligned} &\min_{w,\xi} \rho\|w\|2+\sum_{i}\xi(i) \\ &\quad s.t. \quad \max_{h} wT\Phi(x(i),h,y(i))-\max_{h} wT\Phi(x(i),h,y) \\ &\quad \geq \Delta(y,y(i))-\xi(i) \quad \forall i,\forall y \end{aligned} \tag{3}$$

where i indexes the number of training sample, ρ controls the contribution from the regularization term, and ξ is the slack variable for soft margin formulation. $\Delta(y,y(i))$ measures the cost of misclassifying $y(i)$as y and is defined as

$$\Delta(y,y(i))=\{1 \quad \text{if } y \neq y(i)0\text{otherwise} \tag{4}$$

Eq. (3) can be solved by a non-convex optimization algorithm proposed in [41]. This also generates the training outcome of parameter w which can be used for recognition tasks. In the next subsection, we will discuss in detail how an object is described and how to learn the model.

3. Feature representation strategy

Most big data methods try to simplify the image representation so as to improve the computational efficiency. Such methods, however, often reduce the recognition accuracy. This is conflict with the ultimate goal of studying big data problems which also aims at ensuring accuracy. In our work, we do not focus on extracting simple but limited image features. Instead, feature fusion method is employed to represent each object part and build the part models. This fusion approach combines different kinds of features (e.g., appearance, texture, and shape) and assigns each feature a proper weight. Multiple kernel learning (MKL) method [42] is used to achieve this goal by constructing base kernels for each type of feature. The optimal weights for each base kernel are determined by MKL, and they indicate the contribution/importance of the associated features [43].

We choose four feature types for the feature fusion process, which are appearance feature (SIFT [44]), shape feature (PAS [45]), texture feature (LSS [46]) and color feature (LAB [47]). Using the bag-of-words model [48], these four types of features are quantized into vectors. The vectors are with the same dimension which is 128 in our case (Fig. 1A).

Having got the feature vectors, as described in Fig. 1B, traditional MKL based feature fusion constructs one base kernel for each type of feature. However, it needs to pre-determine the parameters for each kernel, such as the bandwidth σ of Gaussian kernels. If the given parameters are not appropriate, the results would be deteriorated.

To solve the above problem and make the fusion result more robust, we refer to the work of Yeh et al. [43], and use multiple kernels for each feature type (Fig. 1C). Different from [43], all kernels used in this paper are Gaussian kernels, which is proved to be more suitable in our case. Specifically, s equals 10 representing ten different values of σ of Gaussian kernels. Similar to the work of [49], we assign the bandwidth σ to be $\sigma=\{0.5,1,2,5,7,10,12,15,17,20\}$. Thus, each feature type is given ten different Gaussian kernels. By calculating the optimal weights of these kernels, those inappropriate kernels are given small weights and vice versa. So we could get the proper feature fusion results without giving each feature type an exact kernel parameter beforehand.

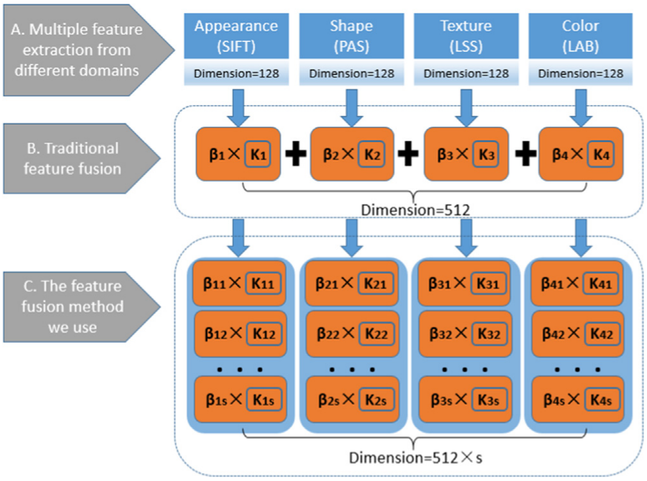


Fig. 1. Illustration of the MKL based feature fusion method proposed in this paper. (A) Four types of features are extracted and quantized into 128 dimensional vectors respectively by using the bag-of-words model. (B) Traditional MKL based feature fusion method constructs one base kernel for each type of feature and concatenates them into a long vector. K_i represents kernels for corresponding features, and β_i is the weight. (C) Our method is an extension of the traditional MKL based feature fusion method. We use s different kernels to represent one feature type. Thus it generates a longer feature fusion vector.

The weights for each kernel are learned through an iterative learning strategy as described in [49]. This iteration is actually an alternative learning process of kernel weight β and Lagrange coefficients α. Each weight β is assigned an equal value $1/(4\times s)$ initially. Next the Lagrange coefficients α can be calculated and then used to update β. This iterative strategy is done in the process of part collection described in Section 4.

Having got the description of the fused features of object and parts, we then analyze the appearance and spatial relationships between parts. The appearance relationship $\psi(hj,hk)$ describes the visual relationship between parts. It can be defined as $\psi(hj,hk)=\sqrt{p(hj)}\times p(hk)$, where $p(hj)$ and $p(hk)$ are the probabilities of being part h_j and h_k. They can be easily calculated using the trained part model.

Meanwhile, the spatial relationship $v(x;j,hj)$ represents the relative distance between part h_j and the object center. It can be defined as follows. First, the Euclidean distance d_j between part h_j and the object center is calculated. Second, the mean value of all the part distances to the object center is calculated: $dc=\frac{1}{m}\sum^{j=1m} dj$, where m is the total part number as illustrated in the beginning of Section 2. Third, to make the relative distance be robust to scale change, the distance d_j is normalized: $dj=dj/dc$.

4. Parts finding and learning process

As part based methods always require much time on image labeling and human annotation for parts, to save human labor, we use only one labeled image for training. However, parts trained on one example tend to perform poorly, we need to improve them by searching for patches that are likely to match. Thus, to maintain training accuracy and get more training samples $\{(x,h,y)\}$, we introduce a part finding algorithm in this section. The algorithm process is illustrated in Fig. 2.

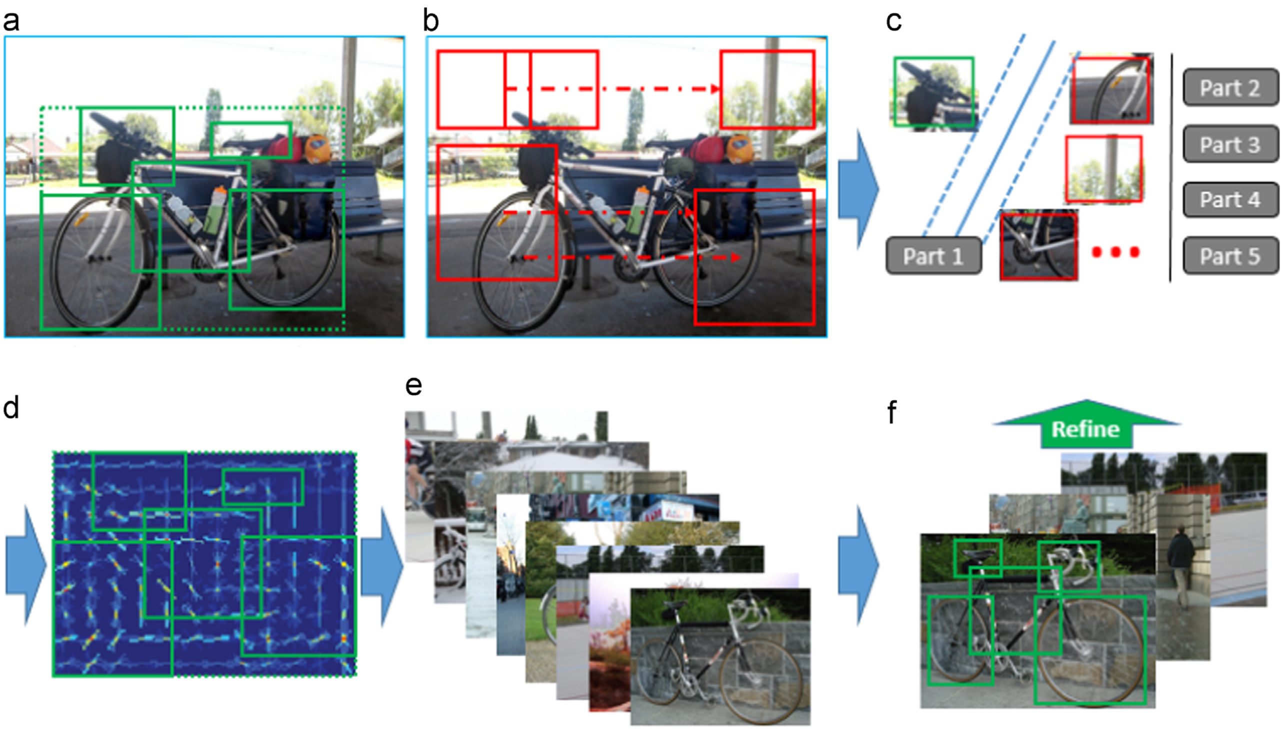


Fig. 2. Illustration of the parts learning and collection process proposed in this paper. (a) A training image with manually located parts is given. (b) For each part, a sliding window method is used to get negative part samples for training. (c) For each part, the part model is trained separately by using Exemplar-SVM method. (d) The structural model is built by integrating the parts as explained in Section 2. (e) Run the structural model on other unlabeled images, and get more candidate positive images for part collection. (f) Select a subset of diverse and discriminative positive candidates, and locate their corresponding parts. Finally, these collected parts are used to refine the part models and the object structural model.

Our method starts by labeling an initial training image. The image should contain at least one positive object. We manually choose the part number and select the parts by drawing rectangular boxes on this image. The areas inside the boxes are considered as the corresponding parts, which is illustrated in Fig. 2a. Note that the rectangular boxes are in different sizes due to different part sizes. The whole object can be located as the smallest box that contains all the parts. It is illustrated as the dotted rectangular box in Fig. 2a.

Next, we use sliding window method to find the corresponding negative parts for each part. The scale size of the window is the same as the corresponding positive part, and the moving step size of the window is set to be 20 pixels. For each positive part, once the windows are collected, we compute the overlap between the windows and the part. To do so, we use the intersection-over-union (IoU) ratio to measure their overlapping area, which is calculated by $Bp\bigcap Bw/Bp\bigcup Bw$. Here B_p and B_w represent the groundtruth window and the detected window respectively. A window is collected as the negative sample of a part if the score of IoU is less than 20%.

Both the positive and negative parts are represented by MKL based multiple feature fusion as described in Section 3. As no iteration has been done yet, the weight value β of the cell features are set to $1/(4\times s)$ initially. s equals 10 representing ten different values of σ of Gaussian kernels (Section 3).

Having got a positive part and several negative ones, we use them to train the part model. Here we use the exemplar-SVM method proposed in [50]. It trains a linear SVM classifier for each exemplar with a single positive example and many negative windows. For a positive part h_i, let N_i denote the set of negative samples for this part. Similar to regular SVM problems, the exemplar-SVM model (ai,bi) can be calculated by solving the following convex objective:

$$\Omega(ai,bi)=\|ai\|2+C1\tau(a^{iT}hi+bi)+C2\sum_{h\in Ni}\tau(-a^{iT}h-bi) \qquad (5)$$

In our method, the regularization parameter C_1 and C_2 are set to 0.5 and 0.01 separately for all part models. The hinge loss function are defined as $\tau(x)=\max(1-x,0)$. The above problem can be solved using LIBSVM tools [51].

For each positive part h_i, we can train an exemplar-SVM model (ai,bi) through the above process. It makes the separate parts available to form the structural model proposed in (1). Furthermore, by calculating the appearance and spatial relationships between these parts, we can get the initial structural model of the given object (Fig. 2d).

Next, we use the initial trained model to get more candidate parts from unlabeled training images for incremental learning. For each unlabeled training image, the initial model can estimate an optimal value of $\Phi(x,h,y)$ by solving (2) and get the optimal location assemble h. Common part based methods often use sliding window method to get the optimal part locations, which is proved to be accurate but time consuming. In this paper, we instead use cross-correlation strategy [52] to improve the efficiency of location estimation. The length-width ratios of the cross-correlation parts are the same with the initial parts, and the scale ratios are set to 5 different values, which are 0.5, 0.8, 1, 1.2, 1.5 times the initial parts.

Having calculated the $\Phi(x,h,y)$ values of all training images, we treat those with higher scores as object candidates which can be used to collect parts and update the structural model. However, these images may share too much common features with the original labeled image. To avoid refining redundant sampled parts candidates, we propose a method to select a small subset of images that are not only complementary but also discriminative.

It is well-known that entropy can be used to measure the uncertainty of an object. High entropy indicates high uncertainty of an image, which in turn suggests possible new information to enrich the model. Inspired by Shannon's entropy theory, we use the following entropy calculating equation to choose a diverse set of images:

$$E(x) = -p(x)\ln p(x) - (1-p(x))\ln(1-p(x)) \tag{6}$$

where $p(x) = 1/[1+\exp(-\Phi(x,h,y))]$ is a process that normalizes the matching score $\Phi(x,h,y)$ of image x to $(0,1)$. Thus, the unlabeled training images with both high matching score $\Phi(x,h,y)$ and high entropy score $E(x)$ are chosen for part collection and structural model updating.

According to the above criteria, more images are chosen to produce parts using the calculated h value. As described in (2), h denotes the optimal location of the part assemble. Meanwhile, it also contains the scale information of these parts. Thus, we could get a diverse set of parts through the above process.

Finally, these chosen parts and images are used to update the initial structural model. There are two main steps. First, the new parts and images are re-described by updating the kernel weight of the MKL feature. Then both the object and the part models are updated. Here the models are built by using regular SVM instead of exemplar SVM because more positive exemplars have been found for training. Second, the appearance and spatial relationships between parts should be updated. The updating implementation details of the two steps can be found in Section 3.

Normally, the part collection process is accomplished when the object model is updated. However, we find the collection results are not good enough because insufficient information is contained in a single training image. Thus we run the part collection process on the training dataset again after a new object model is built. This process is an iterative step if we do not stop it. The experiments in the next section will give detailed evaluation of this iterative process. Furthermore, the model validation for recognition tasks can also be found in the next section.

5. Experimental results

In this section, we validate the effectiveness of the proposed part collection method and the structural model. Firstly, we evaluate the performance of our part collection process and show how we can achieve good results by using incremental learning with different iteration number. Meanwhile, the time complexity is analyzed through the incremental process (Section 5.1). Secondly, we show how accurate our structural model is when dealing with big data recognition tasks. Our method is compared with other structural methods in classification and detection. The results are better than or comparable with alternative methods although only one labeled sample is used for initial model construction. (Section 5.2).

The experiments are performed on three widely used datasets, i.e, Caltech-256 [4], Pascal VOC 2010 [5] and ImageNet datasets [6]. The Caltech-256 dataset contains 30,607 images in 256 categories, with each class containing at least 80 images. The Pascal VOC 2010 dataset consists of 23,374 annotated objects from 10,103 images of 20 classes. We use the ILSVRC'10 subset of ImageNet which contains 1000 classes with more than 1.2 million images. Objects in these three datasets are with a high degree of variation in viewing angle, illumination and object appearance. The Caltech-256 and ImageNet datasets are used for classification performance evaluation while the Pascal VOC 2010 dataset is used for detection performance assessment. All experiments are conducted on a desktop with an Intel Core 2 Duo 2.40-GHz processor without any parallelization.

5.1. Performance of part learning

In the first experiment, we analyze the part finding process proposed in this paper. As described in the previous section, part finding is a key step to train a good model. To evaluate this process conveniently, we saved the part finding results in the process of model updating. Fig. 3 displays some examples of the bicycle part finding results.

It can be seen in this figure that our part finding strategy is effective despite of some bad results. The MKL feature description provides discriminative information of different parts. It ensures the basic accuracy of part finding results. Furthermore, the structural model can provide additional spatial information to help finding parts. If one or two parts are missing or occluded, the structural model can still find their locations if other parts are located correctly. Although bad results are unavoidable, the later incremental learning process can help remove some of them.

Next, we evaluate the time cost of the proposed model in big data application. By analyzing the computing time of this process, we can not only evaluate the algorithm efficiency but also determine when to stop the incremental process. This experiment was done on the Caltech-256 dataset. For each category, we used 30 unlabeled images for training. By annotating the parts of one image initially, we iteratively detect parts in the 30 training images and updated the part model. The training costs are shown in red curve in Fig. 4. To give the effectiveness analysis, we also test the trained models under different iteration numbers on the testing image dataset and give the classification accuracy (the blue curve).

Through the red curve of this figure, we can see the first iteration takes much more time than the following iterations. This is because four types of features are extracted in this iteration. After this step, the time consumption increases almost linearly. Therefore, the most time consuming step of the proposed model is feature extraction. As we need to extract four types of features to construct the MKL feature, it takes five seconds per image by average. The other steps such as feature description, model updating and part finding do not take much time, which is about one second to classify one image. Hence, in big data application, if the feature extraction process has been done and saved in advance, the proposed method can lead to both effective and efficient classification.

Moreover, we can see that the classification accuracy does not always increase when the number of iteration increases. However, the time consumption keeps increasing linearly. Hence, the incremental learning iteration should be stopped on a step to guarantee both efficiency and accuracy. In the following experiments, we set 4 as the iteration number of the incremental learning process.

5.2. Model validation

Now we evaluate the performance of the proposed model for image classification and detection. We also compare it with several state-of-the-art recognition methods. As the testing dataset is with a high degree of view angle variation, it is often difficult for a single structure based model to handle. In our experiment, we build two models for each object category according to different views, which is front view and side view of an object respectively. We chose the model with the highest matching score as the real object model.

First, we evaluate our model on object classification tasks. The experiment is done on the Caltech-256 dataset. Here we choose 5, 15, and 30 as the number of unlabeled training images respectively. Our method is compared with three widely referenced classification methods [53–55]. The method presented in [53] partitions an image into increasingly fine sub-regions and

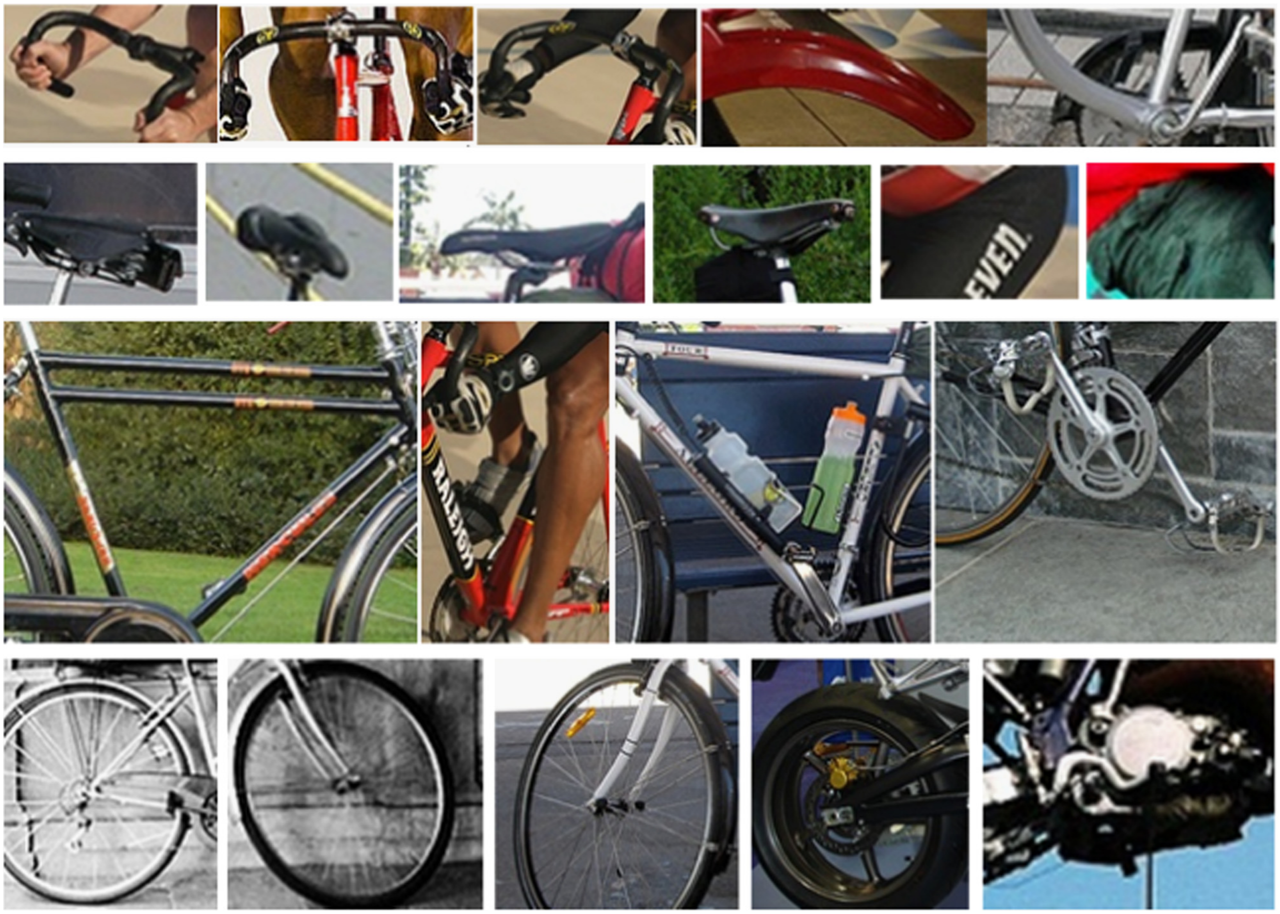


Fig. 3. Some part finding examples from the bicycle category in Pascal VOC 2010 dataset.

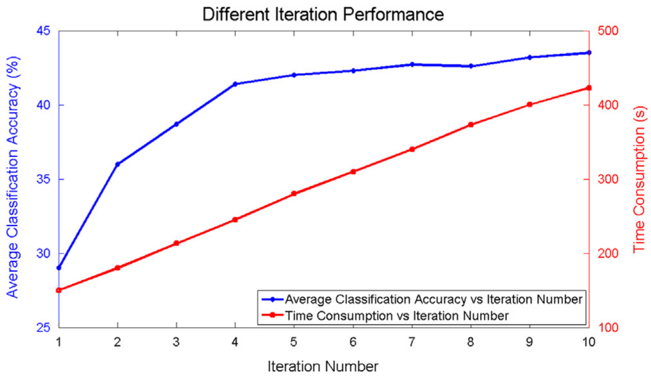


Fig. 4. Performance of the proposed model learning process on the Caltech-256 dataset. The horizontal axis represents the number of incremental learning iterations. The vertical axis on the left is the average classification accuracy of all 256 categories on the testing set. The vertical axis on the right corresponds to the average time consumption over the iteration number on the training set.

generates histograms of local features found inside each subregion. Yang et al. [54] developed an extension of [53] by generalizing vector quantization to sparse coding followed by multiscale spatial max pooling, and proposed a linear SPM kernel based on SIFT sparse codes. Locality-constrained linear coding (LLC [55]) utilizes the locality constraints to project each descriptor into its local-coordinate system, and then generates the final image representation by max pooling.

The classification performance is evaluated using the Average Precision (AP) measure. It computes the area under the Precision/Recall curve, in which higher score means better performance. The classification results are shown in Table 1.

Table 1
Image classification results on Caltech-256 dataset.

Number of training data	5 training	15 training	30 training
Lazebnik et al. [53]	18.40	28.30	34.10
ScSPM [54]	–	27.73	34.02
LLC [55]	–	34.36	41.19
Proposed method	**29.45**	**36.34**	**42.01**

Table 2
Image classification results on ImageNet dataset.

Algorithms	Time consumption (s)	Average precision (%)
Lazebnik et al. [53]	1.40	17.23
ScSPM [54]	1.17	23.65
LLC [55]	**1.08**	30.11
Proposed method	1.18	**56.20**

Through this table, we can see that our method leads the results on all three settings with different numbers of training samples. Although we use only one labeled image for training, the part finding strategy and the incremental framework ensure the overall accuracy of the classification results. When dealing with images with missing or blocked parts, the results show that most

Table 3
Detection results on Pascal VOC 2010 dataset. AP measure (%) is shown for each category.

Category	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow
DPM [28]	**48.7**	52.0	8.9	12.9	32.9	**51.5**	47.1	29.0	13.8	**23.0**
Poselet [56]	33.2	51.9	8.5	8.2	**34.8**	39.0	48.8	22.2	-	20.6
Endres et al. [31]	44.3	35.2	**9.7**	10.1	15.1	44.6	32.0	**35.3**	4.4	17.5
Proposed method	45.1	**52.2**	9.3	**13.5**	26.4	40.2	**51.0**	27.5	**15.3**	20.2

Category	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
DPM [28]	11.1	17.6	42.1	49.3	45.2	7.4	**30.8**	17.1	**40.6**	35.1
Poselet [56]	-	18.5	**48.2**	44.1	**48.5**	9.1	28.0	13.0	22.5	33.0
Endres et al. [31]	**15.0**	**27.6**	36.2	42.1	30.0	5.0	13.7	**18.8**	34.4	28.6
Proposed method	7.8	20.4	37.8	**50.0**	37.6	**12.7**	25.3	15.6	30.8	**37.4**

of them can also be classified correctly due to the superior object description capability of the structural model.

To guarantee the efficiency and effectiveness of our method in dealing with big data problems, we performed experiments on ImageNet dataset which is much larger than the Caltech-256 and the Pascal VOC 2010 datasets. The ILSVRC'10 subset of ImageNet was used and 10 examples from each of the 1000 classes were randomly sampled to build a training set. The remaining examples are used for testing. We followed a one-class classification strategy so that for each class, only data from this class is chosen during the training step. In Table 2, we list the average classification time consumption for one image and the average precision for all classes. Through this table, it can be seen that our method achieves significant higher classification accuracy over the others, which proves the effectiveness of our method in big data problems. Although the proposed method is not the most efficient, its time consumption is comparable to that of the other methods.

Next, we evaluate our model on object detection tasks. The experiment was done on the Pascal VOC 2010 dataset in which the training samples were given by the Pascal 2010 training dataset. We used only one labeled image for initial training, and other samples for incremental training. We compared our method with three part based methods [28,56,31]. In [28], Felzenszwalb et al. proposed one of the most classic methods for Pascal VOC detection task. This method uses latent deformable part model for object detection. The method in [56] discovers poselets that correspond to different parts of human body to detect human pose and other objects. In a more recent work [31], a diverse collection of discriminative parts are learned from object bounding box annotations, and are used to construct part models.

To guarantee consistency with PASCAL detection criterion, a detection is considered correct when it overlaps more than 50% with a ground truth bounding box. Table 3 summarizes the AP for all 20 categories in this dataset.

As shown in Table 3, the proposed method has achieved the highest AP score in 7 categories, and the second highest score in 3 categories. Among all the four part-based models, our method shows better detection accuracy. This proves that the MKL feature description provides more appearance information of the parts, which in turn makes the part models more discriminative between each other. Because our image descriptor is rather high-dimensional but applicable to linear classifiers, it is expected to be more effective when used for large-scale area. On the other hand, the categories with higher scores share the common property that they all have clearly predictable visual properties, such as distinctive parts and relatively fixed spatial arrangement of parts. Thus our structured part model can effectively characterize these properties and achieve good detection performance. For those categories with significant variation in structure, such as *e.g. table* and *bird*, the proposed method is less effective than other part models could be.

6. Conclusion

In this paper we have proposed a new latent structural model that can be used for big data vision problems. The model uses both parts themselves and the relationships (spatial relationship and appearance relationship) between parts to describe objects. This part model is further extended by using MKL features built from more than one kind of feature, which ensures the recognition accuracy. Our method builds an initial object model using only one labeled training image. An incremental learning strategy is employed to find more part candidates and update the model iteratively using an unlabeled training set. Experiments have been performed on three benchmark datasets. The results show that the proposed methods is both effective and efficient in classification and detection tasks, and has outperformed several alternative methods. In the future work, we will investigate other feature descriptions that can make the method more efficient.

Acknowledgments

This work was supported by the NSFC projects (No. 61370123 and 61105002), the Australian Research Councils DECRA Projects funding scheme (project ID DE120102948) and State Key Laboratory of Software Development Environment SKLSDE-2012ZX-22.

References

[1] M.K. Hasan, C.J. Pal, Creating a big data resource from the faces of Wikipedia, in: BigVision, 2012.
[2] B. Gong, F. Sha, K. Grauman, Overcoming dataset bias: an unsupervised domain adaptation approach, in: NIPS Workshop on Large Scale Visual Recognition and Retrieval, 2012.
[3] Z. Wang, R. Liu, Semi-supervised learning for large scale image cosegmentation, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 393–400.
[4] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset. www.vision.caltech.edu/Image_Datasets/Caltech101/.
[5] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes challenge 2010 results, 〈http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html〉.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[7] M. Villegas, R. Paredes, A k-NN approach for scalable image annotation using general web data, in: BigVision, 2012.

[8] D. Tsai, Y. Jing, Y. Liu, H.A. Rowley, S. Ioffe, J.M. Rehg, Large-scale image annotation using visual synset, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 611–618.

[9] Z. Feng, R. Jin, A. Jain, Large-scale image annotation by efficient and robust kernel metric learning, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1609–1616.

[10] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing, 2012.

[11] S.J. Hwang, K. Grauman, F. Sha, Semantic kernel forests from multiple taxonomies, in: BigVision, 2012.

[12] T. Gao, D. Koller, Discriminative learning of relaxed hierarchy for large-scale visual recognition, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2072–2079.

[13] Y. Pang, S. Wang, Y. Yuan, Learning regularized LDA by clustering, IEEE Trans. Neural Netw. Learn. Syst. 25 (12) (2014) 2191–2201.

[14] X. Li, Y. Pang, Deterministic column-based matrix decomposition, IEEE Trans. Knowl. Data Eng. 22 (1) (2010) 145–149.

[15] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, 2004, pp. 1–2.

[16] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[17] Y. Pang, K. Zhang, Y. Yuan, K. Wang, Distributed object detection with linear svms, IEEE Trans. Cybern. 44 (11) (2014) 2122–2133.

[18] Y. Pang, W. Li, Y. Yuan, J. Pan, Fully affine invariant SURF for image matching, Neurocomputing 85 (2012) 6–10.

[19] M.-M. Cheng, Z. Zhang, W.-Y. Lin, P. Torr, Bing: binarized normed gradients for objectness estimation at 300 fps, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[20] D. Zhao, L. Shao, X. Zhen, Y. Liu, Combining appearance and structural features for human action recognition, Neurocomputing 113 (2013) 88–96.

[21] J. Liu, Y. Huang, L. Wang, S. Wu, Hierarchical feature coding for image classification, Neurocomputing 144 (2014) 509–515.

[22] Y. Pang, Z. Ji, P. Jing, X. Li, Ranking graph embedding for learning to rerank, IEEE Trans. Neural Netw. Learn. Syst. 24 (8) (2013) 1292–1303.

[23] Y. Pang, Y. Yuan, K. Wang, Learning optimal spatial filters by discriminant analysis for brain–computer-interface, Neurocomputing 77 (1) (2012) 20–27.

[24] J. Pan, Y. Pang, K. Zhang, Y. Yuan, K. Wang, Energy-saving object detection by efficiently rejecting a set of neighboring sub-images, Signal Process. 93 (8) (2013) 2205–2211.

[25] Y. Lee, K. Grauman, Object-graphs for context-aware visual category discovery, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2) (2012) 346–358.

[26] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3166–3173.

[27] X. Zhang, C. Liu, Image annotation based on feature fusion and semantic similarity, Neurocomputing 149 (2015) 1658–1671.

[28] 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.

[29] P. Ott, M. Everingham, Shared parts for deformable part-based models, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1513–1520.

[30] S. Singh, A. Gupta, A.A. Efros, Unsupervised discovery of mid-level discriminative patches, in: Computer Vision–ECCV 2012, Springer, Oregon, 2012, pp. 73–86.

[31] I. Endres, K.J. Shih, J. Jiaa, D. Hoiem, Learning collections of part models for object recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 939–946.

[32] S. Maji, G. Shakhnarovich, Part discovery from partial correspondence, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 931–938.

[33] H. Zhang, X. Bai, J. Zhou, J. Cheng, H. Zhao, Object detection via structural feature selection and shape model, IEEE Trans. Image Process. 22 (12) (2013) 4984–4995.

[34] F. Zhu, L. Shao, Weakly-supervised cross-domain dictionary learning for visual recognition, Int. J. Comput. Vis. 109 (1–2) (2014) 42–59.

[35] J. Zheng, F. Shen, H. Fan, J. Zhao, An online incremental learning support vector machine for large-scale data, Neural Comput. Appl. 22 (5) (2013) 1023–1035.

[36] C. Chen, X.-W. Wu, B. Sun, J. He, Large-scale image denoising using incremental learning method, Computer, Intell. Comput. Educ. Technol. (2014) 417.

[37] L. Makili, J. Vega, S. Dormido-Canto, Incremental support vector machines for fast reliable image recognition, Fusion Eng. Des. 88 (6) (2013) 1170–1173.

[38] Y. Pang, J. Deng, Y. Yuan, Incremental threshold learning for classifier selection, Neurocomputing 89 (2012) 89–95.

[39] A. Freytag, E. Rodner, P. Bodesheim, J. Denzler, Beyond classification—large-scale Gaussian process inference and uncertainty prediction, in: BigVision, 2012.

[40] Y. Wang, G. Mori, A discriminative latent model of object classes and attributes, in: Computer Vision–ECCV 2010, Springer, Crete, 2010, pp. 155–168.

[41] T.M.T. Do, T. Artiéres, Large margin training for hidden Markov models with partially observed states, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, Montreal, 2009, pp. 265–272.

[42] F.R. Bach, G.R. Lanckriet, M.I. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, in: Proceedings of the 21st International Conference on Machine Learning, ACM, Alberta, 2004, p. 6.

[43] Y.-R. Yeh, T.-C. Lin, Y.-Y. Chung, Y.-C. Wang, A novel multiple kernel learning framework for heterogeneous feature fusion and variable selection, IEEE Transactions on Multimedia 14 (3) (2012) 563–574.

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

[45] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 30 (1) (2008) 0036–51.

[46] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Minnesota, 2007, pp. 1–8.

[47] G. Wyszecki, W. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formulae, Wiley, New York, 1982.

[48] L. Fei-Fei, R. Fergus, A. Torralba, Recognizing and learning object categories, in: ICCV Short Course, 2005.

[49] A. Rakotomamonjy, F.R. Bach, S. Canu, Y. Grandvalet, Simplemkl, J. Mach. Learn. Res. 9 (11) (2008) 2491–2521.

[50] T. Malisiewicz, A. Gupta, A.A. Efros, Ensemble of exemplar-svms for object detection and beyond, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 89–96.

[51] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27, software available at 〈http://www.csie.ntu.edu.tw/cjlin/libsvm〉.

[52] P. Bourke, Cross correlation, in: Auto Correlation-2D Pattern Identification, 1996.

[53] 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.

[54] 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.

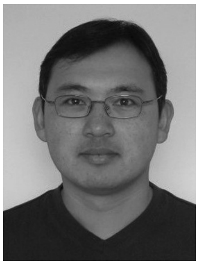
[55] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[56] L. Bourdev, J. Malik, Poselets: body part detectors trained using 3d human pose annotations, in: IEEE International Conference on Computer Vision, 2009, pp. 1365–1372.

Lei Liu was born in Kaifeng, China, in 1978. She received her Bachelor degree in mathematics & applied mathematics, in 2002 and now she is a Ph.D. student in the Department of Mathematics at Shantou University. She is currently a Teacher with the College of Medical at Shantou University. Her research interests include digital image processing, inverse problem and optimization methods.

Xiao Bai received the B.Eng. degree in computer science from Beihang University of China, Beijing, China, in 2001, and the Ph.D. degree from the University of York, York, U.K., in 2006. He was a Research Officer (Fellow, Scientist) in the Computer Science Department, University of Bath, until 2008. He is currently an Associate Professor in the School of Computer Science and Engineering, Beihang University. He has published more than forty papers in journals and refereed conferences. His current research interests include pattern recognition, image processing and remote sensing image analysis. He has been awarded New Century Excellent Talents in University, in 2012.

Huigang Zhang received the Bachelors degree in mathematics from Hebei University of Technology, Tianjin, China, in 2010, and the M.Tech. degree in computer science from Beihang University, Beijing, China, in 2013. His current research interests include structural pattern recognition, statistical machine learning, and image processing.



Jun Zhou received the B.S. degree in computer science and the B.E. degree in international business from Nanjing University of Science and Technology, China, in 1996 and 1998, respectively. He received the M.S. degree in computer science from Concordia University, Canada, in 2002, and the Ph.D. degree in computing science from University of Alberta, Canada, in 2006. He joined the School of Information of Communication Technology at Griffith University, Nathan, Australia, as a Lecturer, in June 2012. Previously, he had been a Research Fellow in the Research School of Computer Science in the Australian National University, Acton, Australia, and a Researcher in the Canberra Research Laboratory, NICTA, Canberra, Australia. His research interests include pattern recognition, computer vision, and machine learning with human in the loop, with their applications to spectral imaging and environmental informatics.

Wenzhong Tang received the Ph.D. degree in Computer Science from Beihang University of China, Beijing, China, in 2008. He is a full Professor in the School of Computer Science and Engineering, Beihang University. He is also holding the positions as the Dean of the operation Institute of Science and Technology and Assistant to President of Beihang University.