Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Neurocomputing

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

Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition

Fuhao Zou [a], Yunfei Wang [a,*], Yang Yang [b], Ke Zhou [c], Yunpeng Chen [a], Jingkuan Song [d]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[c] Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China
[d] Department of Information Engineering and Computer Science, University of Trento, Trento 38100, Italy

ARTICLE INFO

Article history:
Received 14 November 2013
Received in revised form
9 June 2014
Accepted 11 June 2014
Communicated by Chennai Guest Editor

Keywords:
Logistic regression
Stochastic gradient ascent
3D object recognition
Feature learning

ABSTRACT

With the advance of 3D digitalization techniques, it has produced a large number of digital 3D objects, which are usually present in graph, image or video format. In this paper, we focus on designing a novel feature extraction method towards 2D image of 3D object for recognition task. Motivated by the fact that the responses generated by a classifier for two objects can highly reflect their semantic similarity, we attempt to exploit a set of classifiers to construct feature extraction method. The basic idea is as follows. We first learn a classifier for each class and then combine the outputs of all classifiers as object feature. Due to the label information being considered, the proposed method will be more powerful than the typical methods, such as SIFT based bag-of-feature and sparse coding, in terms of discovering the latent semantic information. This is helpful to improve the accuracy of the object recognition. In addition, to make the proposed method scalable to be trained over the massive data (so as to better its generalization ability), the ℓ_2-norm logistic regression is selected as the classifier and trained with stochastic gradient ascent. At the aspect of time complexity, the proposed method is linear to the number of image pixels and less expensive than the other two methods. These arguments have been demonstrated by the obtained experimental results, which is performed over four 3D datasets, such as COIL-100, 3Ddata, ETH-80 and RGB-D dataset.

1. Introduction

With the rapid development of 3D modeling as well as 3D digital image/video capturing, we have witnessed the exponential growth of 3D digital content, such as 3D graph and 3D image and 3D TV/movie [21,30]. Due to the fact that the 3D digital works are able to bring us more vivid and lively vision experience than 2D ones, the investigation related to 3D digital content has attracted a lot of attention in the multimedia community, such as semantic analysis [34,32], scene understanding retrieval [11,6] and recognition [33,7] for 3D objects. As is well known, the feature representation of 3D digital objects plays a fundamental role in the case of multimedia analysis and understanding. Thus, it is highly worthwhile to conduct investigations of how to extract discriminant features for 3D objects. For the purpose of simplifying the problem to be discussed, we mainly concentrate on extracting features for 2D images of 3D objects here.

In principle, the features are roughly grouped into three classes: low level features, middle level features and top level features. Generally, the low level features are built on the low level information of the 3D objects, i.e., the texture information [27,19,12], shape [30,4], color moments [25], Hu's moments invariants [25] and so on. In addition, according to whether or not the interested region of the feature locally or globally corresponds to the image, the low level features are also classified into local features and global features. Most local features represent texture in an image patch. For example, SIFT features use histograms of gradient orientations [19] of the local patch. Global features are composed of contour representations [28], shape descriptors [4], and texture features [27]. Totally, the local or global features intend to capture the distinct features of 3D objects and simultaneously resist the geometrical and photometrical distortion such as translation, rotation, scale, occlusion, clutter and illumination changes.

Though the local features offer the robustness virtues, they are handcrafted and susceptible to suffer the "*semantic gap*" issue. Namely, the low level feature cannot accurately match its top level semantic information. This will result in the fact that the similar objects are far apart in its low level features space with higher probability, which will significantly degrade the performance

* Corresponding author.
E-mail address: yunfeiwang@hust.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.neucom.2014.06.089
0925-2312/

of the 3D object recognition. To handle these issues, some researches attempt to extract the middle or even high level feature via resorting to machine learning methods. For instance, to extract middle level features, sparse coding [36,35] is introduced to build part-based features for 3D object. Suppose that a 3D object to be processed is human, sparse coding can successfully decompose the object as head, leg, and foot body parts by the constrained matrix factorization. In addition, to extract semantic information of the higher level, recently, the deep learning method [26,21,23] has been exploited to extract features for 3D objects.

Compared with the low level feature extraction approaches, the sparse coding and deep learning methods can automatically extract the conceptual and semantic level features respectively. However, they have their own shortcomings. For example, sparse coding is hard to be scalable to large-scale dataset. As to the deep learning methods, training their model is very time-consuming. Apart from this, it requires to be trained by skillful researchers. That is, the performance of the obtained deep learning model is highly dependent on the skills of the researcher. It follows that either sparse coding or deep learning fails to be widely applied to massive data. In other words, handling massive data is a challenging task for the two aforementioned feature learning methods.

To avoid the limitations of the two aforementioned feature learning methods, we intend to propose a fast and scalable feature learning method via a set of one-vs.-all logistic regression classifiers with ℓ_2-norm, also called the ℓ_2-norm logistic regression (abbreviated as ℓ_2-LR). Its motivations are as follows. According to the literature [29], the classifier response values of the input features can be used to measure the similarity among the input features. Therefore, the response values of two similar 3D objects is very close, vise versa. However, if only using a single classifier, it may perform well in the case of measuring the similarity for the similar 3D objects, but fails to recognize dissimilar ones fallen into different categories. For the sake of comprehensively measuring the similarity for various kinds of 3D object, we plan to learn a set of classifiers for all classes of training data and combine their response values into the feature of 3D objects. In addition, to make the feature learning method scalable to massive data, we exploit stochastic gradient ascent to update the model parameters, which is a representative scalable machine learning method.

It is worth highlighting the characteristics of this paper as follows:

- The label information is taken into account when training the feature learning model. This will effectively avoid the semantic gap issue often suffered by the handcrafted feature extraction method.
- Stochastic gradient ascent updating is utilized, which is helpful to training over the massive training data. This implies that we can better the generalization ability via increasing the volume of training data.
- The proposed feature learning method allows us to extract features for test objects in an online way. In contrast, either sparse coding or deep learning fails to achieve this goal, for that the sparse code of the test object is obtained by repeated iterations and deep learning model is too computationally expensive.

The remainder of this paper is organized as follows. In Section 2, we review the state-of-the-art of the feature extraction methods for 3D objects. We present the feature learning algorithm via a set of logistic regression classifiers in Section 3. And then, in Section 4, the extensive experiments are conducted to demonstrate the effectiveness and the efficiency of the proposed algorithm. Finally, we draw a conclusion in Section 5.

2. Related works

Considering the fact that the typical feature extraction methods of 3D objects have been talked about in the Section 1, for simplicity, we plan to just select the state-of-the-art method out of each class to review, such as SIFT and its variants, sparse coding, along with deep learning. The detailed survey is as follows.

2.1. SIFT and its variants

Similar to the 2D object retrieval and recognition applications, scale-invariant feature transform (SIFT) descriptor [19] is also popular and dominant in 3D object applications [30,18,3,1] since the SIFT feature is more superior to other features in terms of handling intensity, rotation, scale and affine variations. However, matching the similarity between 3D object is very tedious if computing the correspondence in the level of SIFT feature. For the ease of matching, the bag-of-feature (BOF) idea is then applied to aggregate the SIFT features in 3D object retrieval and recognition tasks [8,31,3,5].

In spite of the fact that the BOF method is able to successfully aggregate the local features, it will decrease the discriminant ability to some extent because features are aggregated in a disordered way. To remedy this limitation, a multi-resolution version of BoF has been proposed via combing a series of BOFs of different resolutions, which is also called the pyramid kernel [9]. For further improving the discriminant ability, pyramid kernel is extended to spatial pyramid kernel [14] via taking spatial information into account. With the above-mentioned advantages, the pyramid and spatial pyramid kernel have been applied to 3D object recognition [17,13]. As summarized in Section 1, the low level features, such as SIFT and its variants, are readily to result in semantic gap issues because of being constructed in a handcrafted way and label information not being considered.

2.2. Sparse coding

Sparse coding [15] refers to a class of algorithms for automatically finding succinct representations of data as a (often linear) combination of a few typical atoms (usually called dictionaries or codebooks) learned from data. Given only unlabeled input data, it learns basis functions that capture middle level features (i.e., concept level features) in the data. For example, the very large set of English sentences can be encoded by a small number of symbols (i.e. letters, numbers, punctuation, and spaces) combined in a particular order for a particular sentence, and so a sparse coding for English would be those symbols.

More specifically, given a k-dimensional set of real-numbered input vectors $\vec{x} \in \mathbb{R}^k$, the goal of sparse coding is to find n k-dimensional basis vectors $\vec{b_1}, \ldots, \vec{b_n} \in \mathbb{R}^k$ along with a sparse n-dimensional vector of weights or coefficients $\vec{y} \in \mathbb{R}^n$ for each input vector, such that a linear combination of the basis vectors with proportions given by the coefficients results in a close approximation to the input vector: $\vec{x} \approx \sum_{j=1}^{n} y_j \vec{b}_j$ [15]. Presently, sparse coding has been leveraged to extract conceptual features for 2D images of 3D objects [36,35]. Theoretically, the performance of sparse coding based 3D object recognition and retrieval will outperform those on top of low level features (i.e., SIFT and its variants). In spite of this, its extrapolation is quite time-consuming.

2.3. Deep learning

Deep learning is a set of algorithms in machine learning that attempt to learn layered models of inputs, commonly neural

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089

networks. The layers in such models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. With the help of deep learning, the researchers attempt to learn a representation, which makes it easier to accomplish the tasks of interest (e.g., is this the image of a car?).

The term "*deep learning*" is gradually known to us in the mid-2000s after a publication by Geoffrey Hinton [10], in which Hinton investigates how a multi-layered neural network could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, then using supervised backpropagation for fine-tuning. In recent years, deep learning has became a hot topic in machine learning community and is widely used to learn representations for a variety of digital objects. In literatures [21,23], Hinton et al. propose to use Deep Boltzmann Machines and Deep Belief Nets to construct deep learning model to learn top level representations for 3D objects. For deep learning architecture, it is predictable that the deep learning models can output high quality semantic representations than shallow ones, such as sparse coding. However, training the deep learning model is a challenging job since its performance significantly rely on the trainer's skills. Moreover, the deep learning model usually have tens or hundreds of million model parameters, which make it not applicable to online applications.

3. Feature learning algorithm

3.1. The basic idea of the proposed algorithm

Through the previous analysis, it is easy to see that designing a feature extraction method with properties, such as the good generalization ability, low complexity, and excellent semantic representation ability is quite necessary. In view of that the existing classifiers are less complex in terms of prediction purpose and at same time their prediction values can reflect the semantics similarity of two objects [29], we attempt to design a feature extraction method by combining a set of classifiers. The proposed feature extraction method is illustrated in Fig. 1. From the appearance, the architecture of the proposed method is quite similar to that of neural networks, in which each output unit is linearly or non-linearly combined with all input units. However, they are different from each other in principle, which will be confirmed in the following parts.

For the ease of understanding, we assume that there is a training set containing six classes (such as airplane, bike, car, horse, ship and train) available to us. Taking the airplane object as an example, we can train a classifier $f_i(x), i \in \{1, 2, \ldots, 6\}$ over the training set consisting of airplane and non-airplane objects. With the classifier $f_i(x)$, the objects belonging to the airplane class are

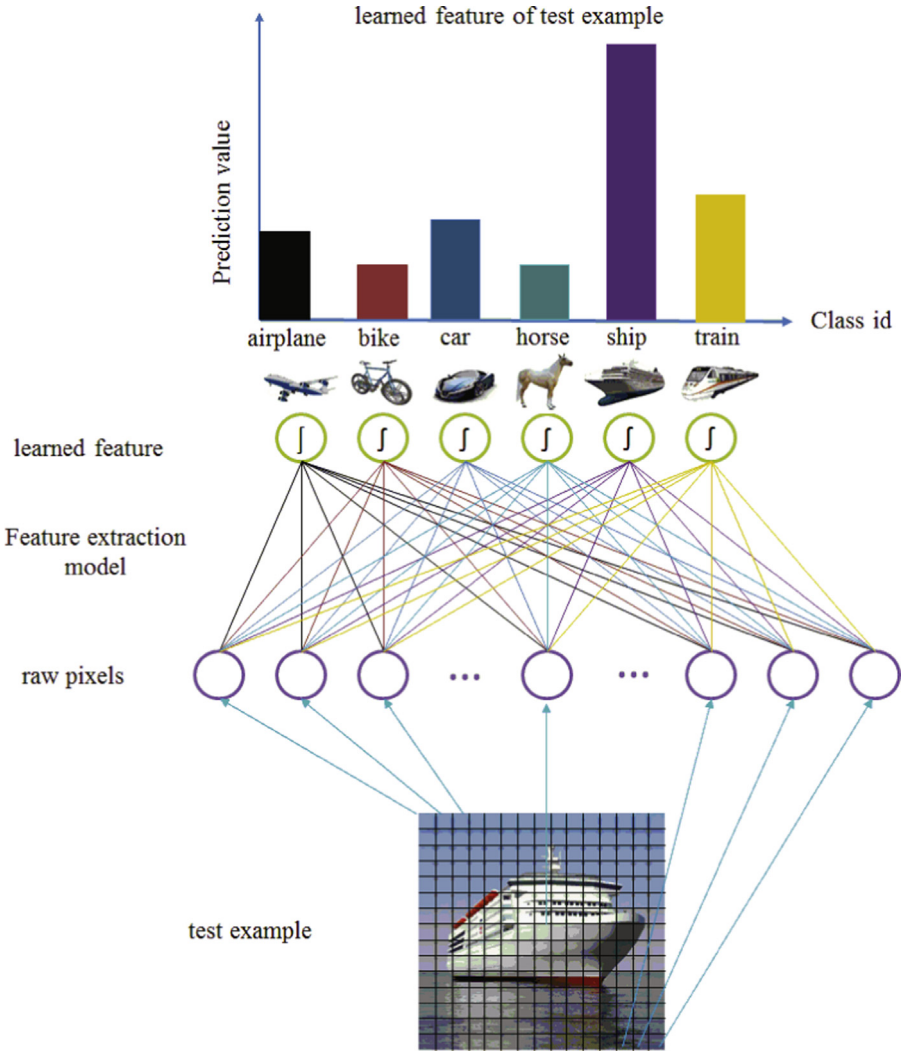


Fig. 1. The framework of the proposed 3D object feature learning model.

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089

associated with similar values and vice versa. However, if only using a single classifier, the prediction values will fails to capture the similarity among the non-airplane objects when they belong to the different classes. To tackle this drawback, we combine the prediction values of the six classifiers as the learned features. Suppose the pixels of 2D image of 3D object are scanned into a vector x in row-wise or column-wise way. Using the proposed feature extraction method, we can obtain its features $y=[y_1,\ldots,y_i,\ldots,y_6]^T$, where $y_i=f_i(x)$.

3.2. Feature learning model

3.2.1. The objective of the proposed model

Though there are a number of binary classifiers (such as support vector machine, logistic regression, and Gaussian discriminant analysis) available to us, here we use ℓ_2-norm regularized logistic regression to construct the feature extraction method. The reason of doing so is that the update rule of the logistic regression with ℓ_2-norm regularization is much easier to be developed into the form of stochastic gradient ascent (SGA) than the other classifiers. As analyzed in literatures [37,29], the update rule of SGA can make the proposed method scalable to large scale dataset for model training. It has shown that simple models and a lot of data trump many elaborate models based on less data. Obviously, we are capable of improving the generalization ability of the proposed method effectively by increasing the volume of the training data.

For the convenience of discussing, we concentrate on how to develop a classifier on top of logistic regression. Assume that the given data is denoted as $\{(x_i,y_i)\}_{i=1}^m$, where $x_i\in\mathbb{R}^n, y_i\in\{0,1\}$. Lets begin with just one training example (x_i,y_i). The logistic regression is represented as

$$f_j(x_i)=g(\beta_j^T x_i)=\frac{1}{1+e^{-\beta_j^T x_i}} \tag{1}$$

where $\beta_j=[\beta_j^0,\beta_j^1,\ldots,\beta_j^n]^T$ and

$$g(z)=\frac{1}{1+e^{-z}}.$$

It follows from Eq. (1) that the probabilities of example x_i belonging to the positive and negative examples are

$$\begin{aligned} P(y_i=1|x_i;\beta_j)&=g(\beta_j^T x_i)\\ P(y_i=0|x_i;\beta_j)&=1-g(\beta_j^T x_i). \end{aligned} \tag{2}$$

To compactly represent it, Eq. (2) can be rewritten as

$$P(y_i|x_i;\beta_j)=(g(\beta_j^T x_i))^{y_i}(1-g(\beta_j^T x_i))^{1-y_i} \tag{3}$$

From the statistical point of view, over the training data $\{(x_i,y_i)\}_{i=1}^m$, the best parameters β_j is usually obtained by maximizing the likelihood quantity

$$\begin{aligned} L(\beta_j) :&= \prod_{i=1}^m P(y_i|x_i;\beta_j)\\ &= \prod_{i=1}^m (g(\beta_j^T x_i))^{y_i}(1-g(\beta_j^T x_i))^{1-y_i} \end{aligned} \tag{4}$$

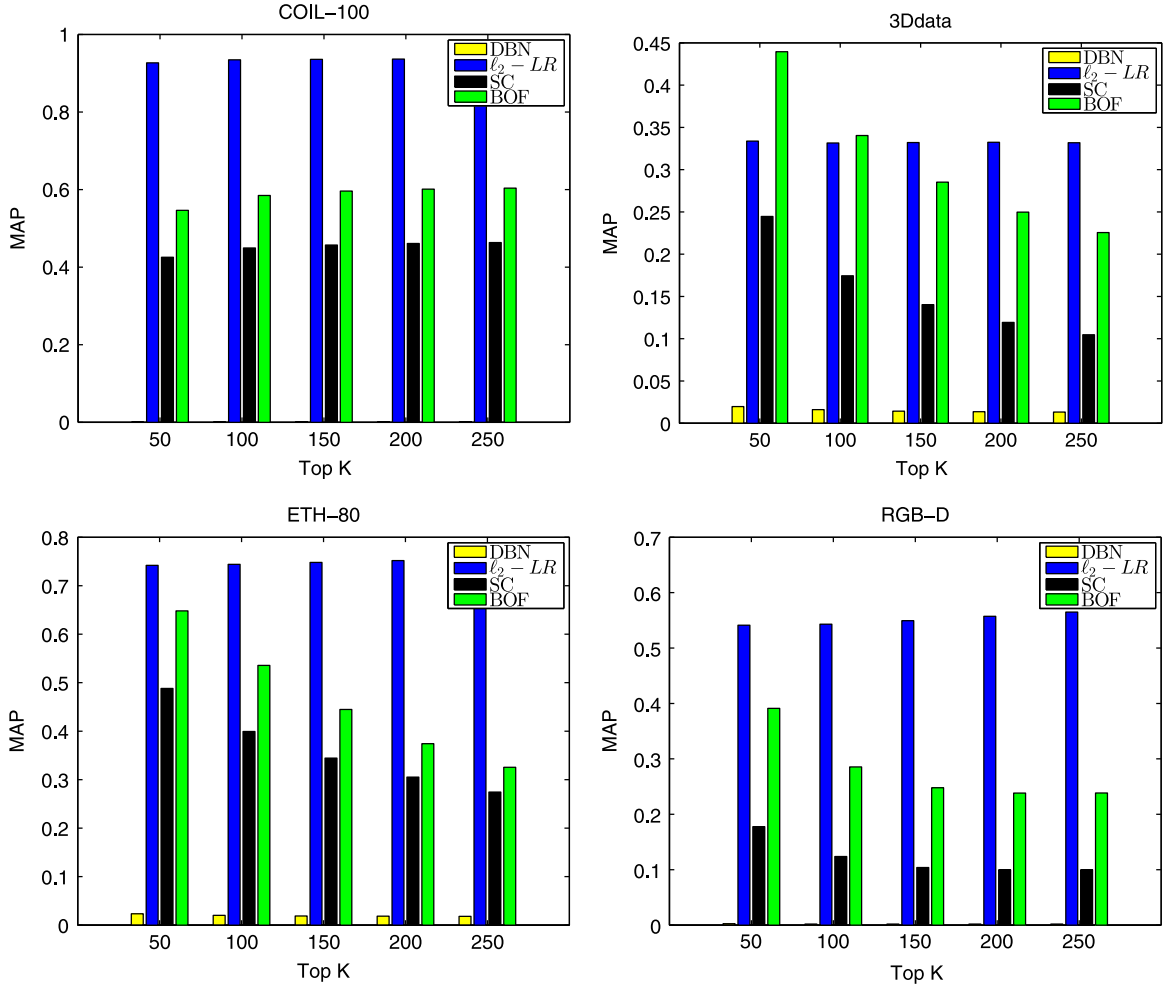


Fig. 2. The MAP of k-nearest neighbors search based object recognition using the features generated by the four selected methods.

As usual, it is more easy to maximize Eq. (4) in its log form:

$$\ell(\beta_j) := \log L(\beta_j) = \sum_{i=1}^{m} y_i \log(g(\beta_j^T x_i)) + (1-y_i)\log(1-g(\beta_j^T x_i)) \quad (5)$$

If just maximizing Eq. (5), it will make the obtained model liable to suffer overfitting issue. To avoid that, the ℓ_2-norm regularization is introduced here. Our objective function can be re-defined as follows:

$$\arg\max_{\beta_j} : \sum_{i=1}^{m} y_i \log(g(\beta_j^T x_i)) + (1-y_i)\log(1-g(\beta_j^T x_i)) - \frac{\lambda}{2}\|\beta_j\|^2 \quad (6)$$

In Eq. (6), the first part measures the degree of the model fitting to the given data and the second part (i.e., regularization item) is used to smooth the model for suppressing overfitting problem.

3.2.2. Updating rules of the learning model

For the purpose of scalability to train over massive data, we intend to derive a stochastic gradient ascent update rule to find the optimal parameters $\beta_j = [\beta_j^0, \ldots, \beta_j^k, \ldots, \beta_j^n]$. To this end, we only consider one training example (x_i, y_i) available. By taking derivative of Eq. (6) with respect to β_j^k, we can obtain the gradient of β_j^k

$$\frac{\partial \ell(\beta_j)}{\partial \beta_j^k} = (y_i - g(\beta_j^T x_i))x_i^k - \lambda\beta_j^k \quad (7)$$

where x_i^k is the kth element of vector x_i. Accordingly, the parameter β_j can update element-by-element in the stochastic gradient ascent way

$$\beta_j^k = \beta_j^k + \alpha((y_i - g(\beta_j^T x_i))x_i^k - \lambda\beta_j^k) \quad (8)$$

where α is the learning rate.

In practice, most of the training tasks are realized in the single processor environment. If so, the proposed algorithm on top of stochastic gradient ascent can be illustrated like (1) when the number of iterations equals T. The iterative procedure is usually terminated when T reaches the upper bound T_{max} or the parameters have little change during several successive iterations. Accordingly, its time and space complexity are $O(mT)$ and $O(m)$ respectively.

Algorithm 1. SingleSGA: ℓ_2-LR on single processor.

Input:
The training set $(x_i, y_i)_{i=1}^m$;
The regularization factor λ;
The learning rate α;
Output:
The model parameter β_j;
1: shuffle the training set $(x_i, y_i)_{i=1}^m$ randomly;
2: **for** $t = 1$ *to* T **do**
3: **for** $i = 1$ *to* m **do**
4: Compute each element of β_j according to equation (8);
5: **end for**
6: **end for**
7: RETURN β_j

3.2.3. Extension to large-scale learning

For most of machine learning tasks, it is desired to devise a powerful model with good generalization ability. However, it is very tough for most of investigators to achieve such goal. In practice, there is an easy but effective way to improve the generalization ability via increasing the volume of the training data. This idea has been realized in a few literature [37,2] by modifying the algorithm from single task mode to parallel processing mode. Especially, with the advance of the distributed and parallel computing (such as cloud computing, GPU computing), it is very convenient to achieve this target. Following this idea, we propose a parallel version of stochastic gradient ascent illustrated in (2).

Algorithm 2. ParallelSGA: ℓ_2-LR on multi-processor.

Input:
The training set $(x_i, y_i)_{i=1}^m$;
The regularization factor λ;
The learning rate α;
The number of computer or core k;
Output:
The model parameter β_j;
1: Set partition size $Z = \lfloor \frac{m}{k} \rfloor$
2: Randomly partition the training data into k subsets, each computer or core is assigned a subset (i.e., Z examples);
3: **for** $i = 1$ *to* k **do**
4: shuffle the training subset on computer or core i;
5: Compute $(\beta_j)^i$ using the (1) over the i-th training subset;
6: **end for**
7: Aggregate from all computers or cores $\beta_j = \frac{1}{k}\sum_{i=1}^{k}(\beta_j)^i$
8: RETURN β_j

4. Experiments

To fairly evaluate the proposed feature extraction method, three kinds of method including SIFT-based BOF, sparse coding (SC) and deep belief networks (DBN) [20] are selected as comparative objects from the three typical feature extraction methods. We conduct a series of experiments to verify the proposed approach's semantic representation ability, efficiency, scalability as well as sensitivity to the varying of the regularization factor λ.

4.1. Dataset description

The experiments are operated on four 2D image of 3D object databases: COIL-100, 3Ddataset, ETH-80 and RGB-D. The short description related to such four databases is as follows:

COIL-100: Columbia Object Image Library (COIL-100) is a database of color images of 100 objects. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360° to vary object pose with respect to a fixed color camera. Images of the objects were taken at pose intervals of 5°. This corresponds to 72 poses per object. The images were size normalized. COIL-100 is available online.[1] For formal documentation look at the corresponding compressed technical report [22].

3Ddataset provided by Feifei Li: The dataset[2] consists of 10 object categories (bicycle, car, cellphone, head, iron, monitor, mouse, shoe, stapler, toaster). For each object category, the dataset contains images of 10 individual object instances under eight viewing angles, three heights and three scales for a total number of approximate 7000 images. Images are roughly 400 × 300 pixels in bmp format [24].

ETH -80: This dataset[3] is the standard version of the ETH-80 database, applicable for almost all experiments. All images are cropped, so that they contain only the object, centered in the image, plus a 20% border area (to avoid border effects when

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php
[2] http://www.eecs.umich.edu/vision/data/3Ddataset.zip
[3] http://www.d2.mpi-inf.mpg.de/Datasets/ETH80

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089

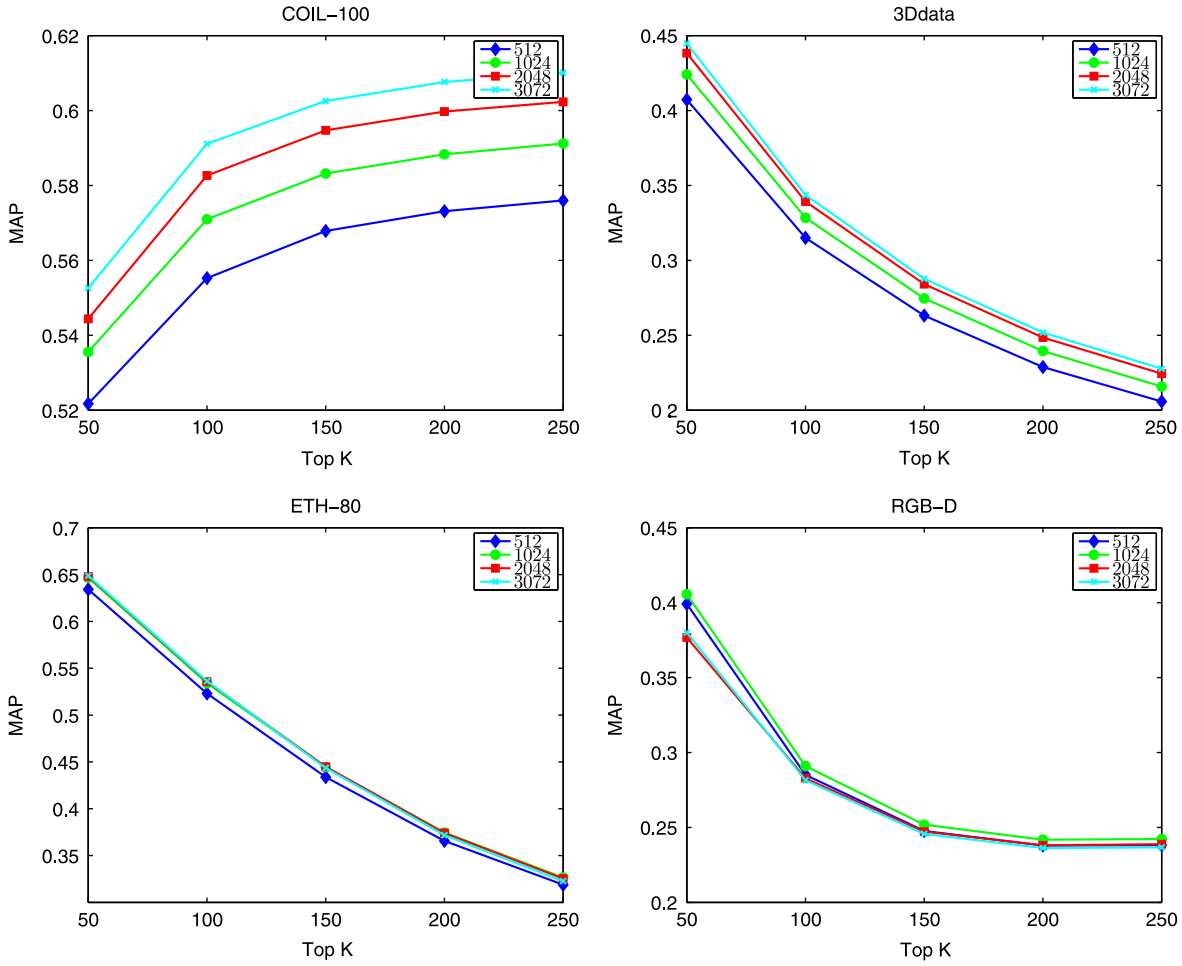


Fig. 3. The MAP of *k*-nearest neighbors search based object recognition using the features generated by different sizes of BOF models.

derivatives need to be calculated). In contrast to the COIL database, the scale is left the same for all images of the same object. Images are rescaled to a size of 256∗256 pixels [16].

RGB-D Dataset: The RGB-D Dataset[4] is a large dataset of 300 common household objects. The objects are organized into 51 categories arranged using WordNet hypernym–hyponym relationships (similar to ImageNet). This dataset was recorded using a Kinect style 3D camera that records synchronized and resized 78×84 RGB. Each object was placed on a turntable and video sequences were captured for one whole rotation. For each object, there are three video sequences, each recorded with the camera mounted at a different height so that the object is viewed from different angles with the horizon.

4.2. Object recognition using k-nearest neighbors search

In order to evaluate the performance of the proposed method, we extract the feature vectors from the four selected datasets and calculate the mean average precision (MAP) value of the object recognition using the *k*-NNS. Intuitively, the feature vectors of the objects with the same classes have the close distance and vice versa. Thus, the more sufficient the features reflect the similarity of the object, the more accurate the precisions of the recognition are. Namely the accuracy of the object recognition can mainly represent the extent of the goodness or badness of the extracted feature.

The experimental setting is as follows. Except the RGB-D Dataset, in which only 10,000 images are randomly selected for test, the objects in other three datasets are all selected for calculating recognition precision. For each selected dataset, about 30% images are randomly selected from each object class as test dataset and the rest are used as a reference dataset. In BOF, 128-dimension SIFT features are extracted from each image and then aggregated as *K*-dimensions feature with the help of a BOW model based on *K*-means. We have tried different sizes ($K = 512, 1024, 2048, 3072$) of BOF model of SIFT features and the corresponding MAP scores are shown in Fig. 3. Intuitively speaking, the BOF model with larger size can capture more details of images. However, according to the MAP scores of BOF, we find that the BOF model with the size of 2048 has a satisfying trade-off between the computation cost and the accuracy on the four datasets. So we set $K = 2048$ in the following experiments for BOF. In ℓ_2-LR, SC and DBN, each image is transformed into a gray-scale image with the size of 80∗60. A row vector formed by these gray-scale values is used as the feature *x* of each image.

For ℓ_2-LR and DBN, the reference dataset is served as the training dataset to generate the feature extract function. In ℓ_2-LR, the learning rate α is adaptively set with line search method and the regularization parameters associated with COIL-100, 3Ddata-set, ETH-80, RGB-D λ are empirically set as $10^{-5}, 10^{-5}, 10^{1}, 10^{-1}$ respectively so as to approximate the best performance. In DBN, there are two hidden layers and the number of hidden units of both hidden layers is 100. When pre-training and fine tuning DBN, the number of full swaps through is 150. Based on these training data, we can learn a codebook served as the basis vectors of a

[4] http://www.cs.washington.edu/rgbd-dataset/

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089

feature space. We utilize a non-negative matrix factorization model with a L2-norm regularization to conduct sparse coding, in which the number of basis vectors is 100 and the parameter associated with the regularization term is 0.01.

Fig. 2 presents the MAP values of the four methods over the selected datasets. From Fig. 2, it can be observed that the ℓ_2-LR is superior to the other three methods among the COIL-100, ETH-80, and RGB-D Datasets and is defeated only in the case that the parameter k of k-NNS is set 50 over the 3Ddataset. In our opinion, the performance gains mainly stems from the fact that the label information is incorporated for generating the feature extraction function. Under the supervision of the label information, the information closely relevant to the specific class is activated and vise versa. Recall that in principle, the sparse coding should be competitive with our method since it can discover the semantic information of the processed object. However, probably the complex backgrounds of test object degrade its performance since the partial background content is also taken into account in sparse coding. This is also why the BOF performs better than the sparse coding in terms of recognition accuracy, for it is robust against a variety of distortions.

Table 1
The average time cost per image (millisecond/per image) associated with BOF, sparse coding (SC), ℓ_2-LR and DBN.

Extraction method	BOF	SC	ℓ_2-LR	DBN
COIL-100	349	4	0.1	0.3
3Ddataset	123	4.1	0.1	0.3
ETH-80	328.7	4.1	0.1	0.3
RGB-D Dataset	94.9	4.5	0.1	0.2

4.3. Time cost of feature extraction

For most of the online applications, they require that the speed of object recognition is as fast as possible. Functionally, object recognition is composed of two processes, such as feature extraction and object classification. This implies the feature extraction method with lower complexity is more preferable for online applications. For BOF, the time cost is mainly spent on extracting the SIFT features and aggregating the SIFT features into BOF. In practice, one image usually corresponds to about hundreds or even thousands of SIFT features. Therefore, even regardless of aggregating process, the BOF will still be time-consuming since extracting the local features is computationally expensive. As to sparse coding, its complexity is also relatively high because the feature of test object is obtained in an iterative way. Compared with the former two methods, the ℓ_2-LR is absolutely faster since its complexity is linear to the number of pixels. This argument has been demonstrated in the following experiment. We use the three selected methods to extract the features over the four datasets and calculate the average time cost of each image. The result is illustrated in Table 1. It is easy to see that the ℓ_2-LR is more efficient.

4.4. Sensitivity analysis of the parameters

There are two parameters α and λ to be tuned in the objective function so as to approximate the optimal performance.

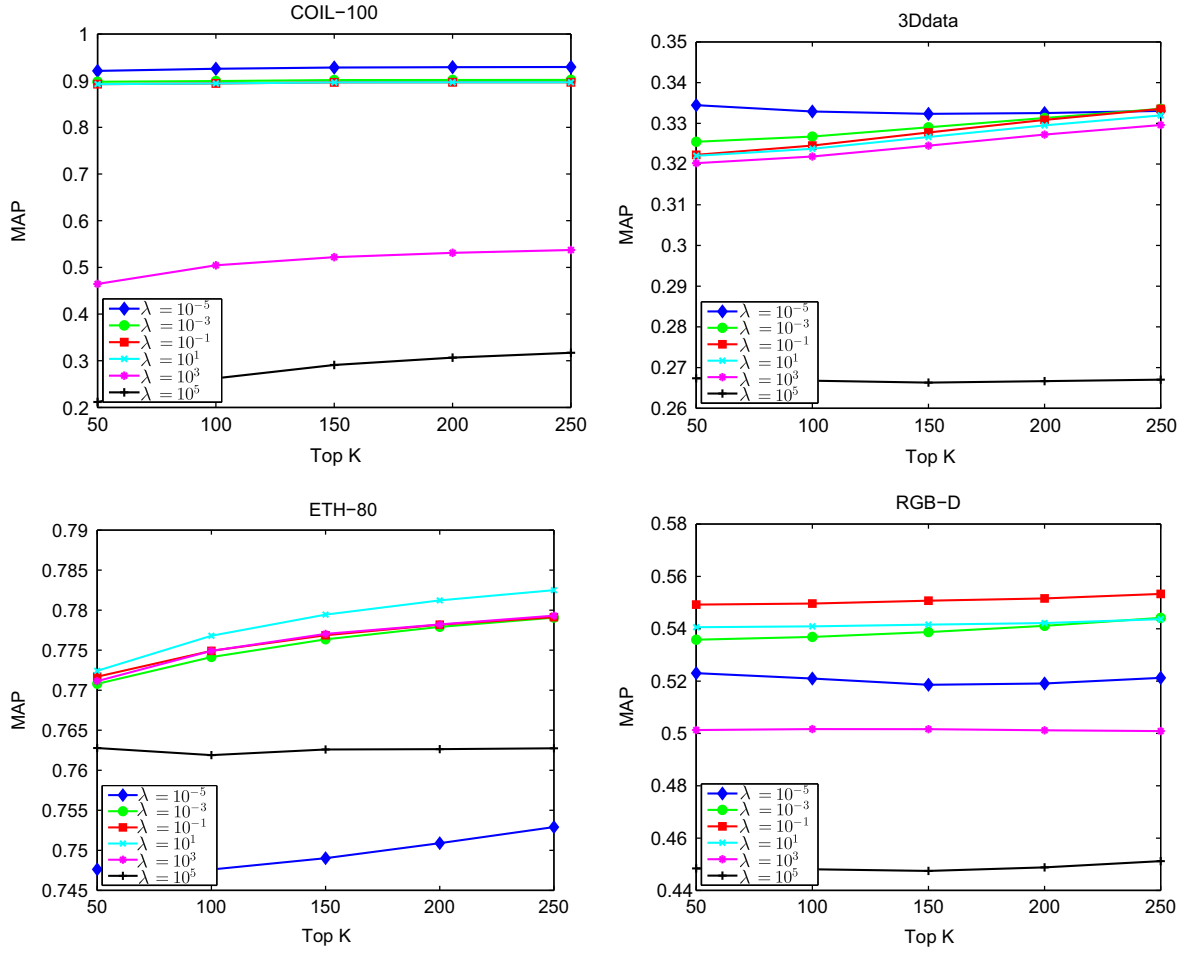


Fig. 4. The MAP values corresponding to different λ over the four selected datasets.

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089

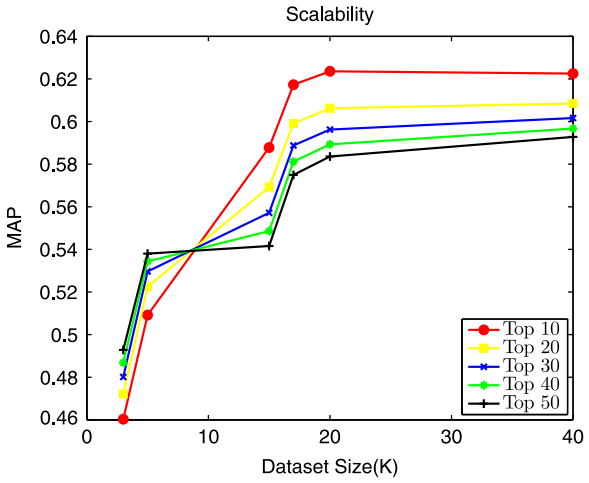


Fig. 5. The MAP values associated with different dataset sizes.

As pointed out in Section 4.2, the parameter α is automatically obtained using line search (e.g., L-BFGS). Then, we will focus on how to set the values for parameter λ. As usual, we adopt the grid search way to tune the parameter λ. We vary the parameter λ in ranges $\{10^{-5}, 10^{-3}, 10^{-1}, 10^{1}, 10^{3}, 10^{5}\}$. Under these settings, we calculate the MAP values with respect to top K relevant documents over the four selected datasets and plot their curves in Fig. 4.

From Fig. 4, we find two phenomena: (1) the ℓ_2-LR is sensitive to the parameter λ; (2) for the parameter λ, it acts as a tuner of the generalization ability. For example, with respect to datasets Coil-100 and 3Ddata, maybe the distribution of the in-sample and out-of-sample is approximately identical and therefore role of regularizer λ can be ignored. But, for the two remaining datasets, we will leverage the parameters λ to reduce the impact of the distribution divergence between in-sample and out-of-sample so as to obtain the optimal generalization ability.

4.5. Scalability to large scale data

For ℓ_2-LR, its update rule is based on a stochastic gradient ascent. This means it can adapt to training over massive datasets, which improves the generalization ability of the ℓ_2-LR. To verify this statement, we select the largest dataset, RGB-D, from the four datasets and calculate its MAP values associated with different data sizes. Here, we vary the data size in the range of $\{3K, 5K, 15K, 17K, 20K, 40K\}$. Fig. 5 presents the experimental results. From Fig. 5, we find that the MAP gradually increases with the increasing of the data size. Obviously, our proposed method is feasible to improve its generalization ability with the help of increasing the data size.

5. Conclusion and future work

In this paper, we propose a new feature extraction method, ℓ_2-LR, to automatically extract feature vectors for 2D images of 3D objects. Due to the fact that label information is taken into account, the resultant features have more powerful semantic representation ability, which is beneficial to offer a higher recognition rate. In contrast to the other two feature extraction methods, such as BOF and sparse coding, the proposed method requires that the label information is available. However, this does not become its barrier to be applied to object recognition since the label information is easy to get by resorting to the existing labeling techniques. In addition, we adopt the stochastic gradient ascent to update our model parameter, which makes the proposed method scalable to train over big data so as to get a powerful model. The experimental results demonstrate that our method is much faster than the other two methods. Besides, it can be competitive or even better compared with the other two methods in terms of recognition accuracy.

Acknowledgments

This work is supported in part by the National Basic Research Program (973 Program) of China under Grant no. 2011CB302305 and the National Natural Science Foundation of China under Grant no. 61232004.

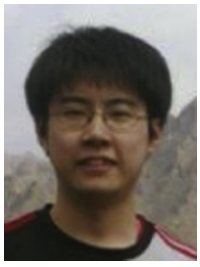
References

[1] M. Brown, D.G. Lowe, Unsupervised 3d object recognition and reconstruction in unordered datasets, in: Fifth International Conference on 3-D Digital Imaging and Modeling, 2005. 3DIM 2005, IEEE, pp. 56–63.
[2] C.T. Chu, S.K. Kim, Y.A. Lin, Y. Yu, G. Bradski, A.Y. Ng, K. Olukotun, Map-reduce for machine learning on multicore, in: NIPS, vol. 6, pp. 281–288.
[3] E.M. Daoudi, C. Tadonki, et al., 3d shape retrieval using bag-of-feature method basing on local codebooks, in: Image and Signal Processing, Springer, 2012, pp. 391–396.
[4] P.E. Forssen, D.G. Lowe, Shape descriptors for maximally stable extremal regions, in: IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, IEEE, pp. 1–8.
[5] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, T.S. Chua, Camera constraint-free view-based 3-d object retrieval, IEEE Trans. Image Process. 21 (2012) 2269–2281.
[6] Y. Gao, M. Wang, R. Ji, X. Wu, Q. Dai, 3-d object retrieval with Hausdorff distance learning, IEEE Trans. Ind. Electron. 61 (2014) 2088–2098.
[7] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (2012) 4290–4303.
[8] Y. Gao, M. Wang, Z.J. Zha, Q. Tian, Q. Dai, N. Zhang, Less is more: efficient 3-d object retrieval with query view selection, IEEE Trans. Multimed. 13 (2011) 1007–1018.
[9] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, in: Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, vol. 2, IEEE, pp. 1458–1465.
[10] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[11] R. Ji, L.Y. Duan, J. Chen, T. Huang, W. Gao, Mining compact 3d patterns for low bit rate mobile visual search, IEEE Trans. Image Process. XX (2014), in press.
[12] R. Ji, H. Yao, W. Liu, X. Sun, Q. Tian, Task-dependent visual-codebook compression, IEEE Trans. Image Process. 21 (2012) 2282–2293.
[13] N. Larios, J. Lin, M. Zhang, D. Lytle, A. Moldenke, L. Shapiro, T. Dietterich, Stacked spatial-pyramid kernel: an object-class recognition method to combine scores from random trees, in: 2011 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, pp. 329–335.
[14] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, pp. 2169–2178.
[15] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, Adv. Neural Inf. Process. Syst. 19 (2007) 801.
[16] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, vol. 2, IEEE, pp. II–409.
[17] X. Li, I. Guskov, 3d object recognition from range images using pyramid matching, in: IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, IEEE, pp. 1–6.
[18] Q. Liu, H. Xiao, Semi-global depth estimation algorithm for mobile 3-d video applications, Tsinghua Sci. Technol. 17 (2012) 128–135.
[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[20] A.r. Mohamed, T.N. Sainath, G. Dahl, B. Ramabhadran, G.E. Hinton, M.A. Picheny, Deep belief networks using discriminative features for phone recognition, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 5060–5063.
[21] V. Nair, G.E. Hinton, 3d object recognition with deep belief nets, in: NIPS, pp. 1339–1347.
[22] S.K. Nayar, S.A. Nene, H. Murase, Columbia object image library (coil 100), Technical Report, Department of Computer Science, Columbia University, CUCS-006-96, 1996.
[23] R. Salakhutdinov, G. Hinton, An efficient learning procedure for deep Boltzmann machines, Neural Comput. 24 (2012) 1967–2006.
[24] S. Savarese, F.F. Li, 3d generic object categorization, localization and pose estimation, in: ICCV, pp. 1–8.
[25] X. Sheng, P. Qi-Cong, 3d object recognition using multi-moment and neural network, in: International Conference on Communications, Circuits and Systems, 2008. ICCCAS 2008, IEEE, pp. 1000–1004.
[26] R. Socher, B. Huval, B.P. Bath, C.D. Manning, A.Y. Ng, Convolutional-recursive deep learning for 3d object classification, in: NIPS, pp. 665–673.

[27] F. Tombari, S. Salti, L. Di Stefano, A combined texture-shape descriptor for enhanced 3d feature matching, in: 2011 18th IEEE International Conference on Image Processing (ICIP), IEEE, pp. 809–812.
[28] C. Urdiales, C. de Trazegnies, J. Pacheco, F. Sandoval, View planning for efficient contour-based 3d object recognition, in: MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference, IEEE, pp. 206–211.
[29] G. Wang, D. Hoiem, D. Forsyth, Learning image similarity from Flickr groups using fast kernel machines, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 2177–2188.
[30] P. Yan, S.M. Khan, M. Shah, 3d model based object class detection in an arbitrary view, in: IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, IEEE, pp. 1–6.
[31] Y. Yang, Y. Yang, Z. Huang, H.T. Shen, F. Nie, Tag localization with spatial correlations and joint group sparsity, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 4, pp. 881–888.
[32] Y. Yang, Y. Yang, H.T. Shen, Effective transfer tagging from image to video, ACM Trans. Multimed. Comput. Commun. Appl. 9 (2013).
[33] Y. Yang, Y. Yang, H.T. Shen, Y. Zhang, X. Du, X. Zhou, Discriminative nonnegative spectral clustering with out-of-sample extension, IEEE Trans. Knowl. Data Eng. 25 (2013) 1760–1771.
[34] Y. Yang, Z.J. Zha, Y. Gao, X. Zhu, T.S. Chua, Exploiting web images for robust semantic video indexing via sample-specific loss, IEEE Trans. Multimed. XX (2014), in press.
[35] X.X. Yin, B.H. Ng, K. Ramamohanarao, D. Abbott, Tensor based sparse decomposition of 3d shape for visual detection of mirror symmetry, Comput. Methods Prog. Biomed. 108 (2012) 629–643.
[36] S.M. Yoon, T. Schreck, G. Yoon, Sparse coding based feature optimisation for robust 3d object retrieval, Electron. Lett. 48 (2012) 493–495.
[37] M. Zinkevich, M. Weimer, A.J. Smola, L. Li, Parallelized stochastic gradient descent, in: NIPS, vol. 4, p. 4.

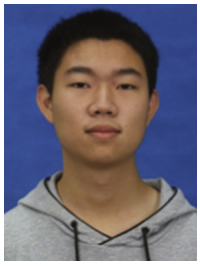
Fuhao Zou received B.E. degree in computer science from Huazhong Normal University, Wuhan, Hubei, China, in 1998. And received M.S. and Ph.D. degrees in computer science and technology from Huazhong University of Science and Technology (HUST), Wuhan, Hubei, China, in 2003 and 2006. Currently, he is an associate professor with the College of Computer Science and Technology, HUST. His research interests include machine learning, multimedia understanding and analysis, big data analysis, semantic based storage, and cloud storage. He is senior member of China Computer Federation (CCF) and member of IEEE, ACM.

Yunfei Wang received the B.S. degree from computer science and technology, Wuhan University of Science and Technology, Wuhan, China. He is currently pursuing the Master's degree at Huazhong University of Science and Technology, Wuhan, China. His current research interests include machine learning and data mining, etc.



Yang Yang received his Ph.D. degree in information technology from the University of Queensland, Australia. He is an associate professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interest includes multimedia information retrieval, social media analytics, computer vision, machine learning, data mining, etc.

Ke Zhou received the B.E., M.E., and Ph.D. degrees in computer science and technology from Huazhong University of Science and Technology (HUST), China, in 1996, 1999, and 2003, respectively. He is a full professor of the School of Computer Science and Technology, HUST. His main research interests include computer architecture, cloud storage, parallel I/O and storage security. He has more than 50 publications in journals and international conferences, including Performance Evaluation, FAST, MSST, ACM MM, SYSTOR, MASCOTS and ICC. He is a member of IEEE.



Yunpeng Chen is currently an undergraduate student in Huazhong University of Science and Technology. He majors in computer science and is supervised by Fuhao Zou. His research interest includes machine learning and its applications to multimedia content analysis and computer vision.

Jingkuan Song received his Ph.D. degree in information technology from the University of Queensland, Australia. He received his B.S. degree in software engineering from the University of Electronic Science and Technology of China. Currently, he a postdoctoral researcher in the Department of Information Engineering and Computer Science, University of Trento, Italy. His research interest includes large-scale multimedia search and machine learning.

Please cite this article as: F. Zou, et al., Supervised feature learning via l_2-norm regularized logistic regression for 3D object recognition, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.089