J. Vis. Commun. Image R. 33 (2015) 368–377

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Image automatic annotation via multi-view deep representation ☆

Yang Yang, Wensheng Zhang *, Yuan Xie

Institute of Automation, University of Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 16 February 2015
Accepted 9 October 2015
Available online 22 October 2015

Keywords:
Image annotation
Stacked auto-encoder
Imbalance learning
Multi-view learning
Image features
Semantic gap
Deep learning
Multi-labeling

ABSTRACT

The performance of text-based image retrieval is highly dependent on the tedious and inefficient manual work. For the purpose of realizing image keywords generated automatically, extensive work has been done in the area of image annotation. However, how to treat image diverse keywords and choose appropriate features are still two difficult problems. To address this challenge, we propose the multi-view stacked auto-encoder (MVSAE) framework to establish the correlations between the low-level visual features and high-level semantic information. In this paper, a new method, which incorporates the keyword frequencies and log-entropy, is presented to address the imbalanced distribution of keywords. In order to utilize the complementarities among diverse visual descriptors, we tactfully apply multi-view learning to search for the label-specific features. Thereafter, the image keywords are finally produced by appropriate features. Conducting extensive experiments on three popular data sets, we demonstrate that our proposed framework can achieve effective and favorable performance for image annotation.

1. Introduction

With the remarkable improvement of the information technic, the proliferation of digital images on the Internet posed a great challenge for large-scale image management. In order to organize, query and scan so large-scale images, image retrieval has been established. Text-based image retrieval (TBIR) [1], a typical image retrieval system, allows a user to present his/her information need as textual query and find the relevant images based on the match between the textual query and the manual annotations of images. Carefully chosen keywords can improve image retrieval accuracy, but the tagging process is known to be tedious and labor intensive [2]. Due to the tagged image databases containing rich information about the semantic meanings of the images, many image annotation algorithms exploit the exist tagged images to automatically annotate new images or add extra keywords to images with a few existing keywords.

The annotation process can be considered as a multi-label classification problem, and the existing methods can be grouped into two categories: generative models and discrimination models. Generative models try to learn the joint probability distribution between semantic concepts and image visual features, thus the image annotation can be achieved by using probabilistic inference [3–5,9,6–8]. Meanwhile, the discrimination models take the image annotation task as a supervised learning problem. Many typical supervised learning models have been introduced into this task, such as hidden Markov model (HMM) [10], supervised multi-class labeling (SML) [11], and support vector machine (SVM) [12,13]. Compared with the previous work, recent research efforts have taken more attention on extending the keyword prior knowledge and using hybrid model [14,18,19,17,15,16]. Recently, deep learning, as a novel machine learning algorithm, has achieved wonderful performance in the field of image understanding. This algorithm is also introduced to image annotation problem [25,28].

Although these algorithms have got wonderful results, the results are still unsatisfied. As a matter of fact, the annotation performance is limited in two major aspects:

(1) In real situations, image keywords from the Internet are extremely diverse with non-uniform distribution. Table 1 shows the image keyword distributions in three data sets, and we can see that the number of different keywords used for label images are seriously imbalanced. In this situation, the learning model trends to produce high accuracy for major keywords and poor performance for minority keywords. The average performance for all keywords of the model is limited by the minority keywords.
(2) Images are often described by diverse features which own different recognition capacities for different objects. For instance, in Fig. 1 "sky" and "sea" are likely identified by color, while "airplane" and "bird" are likely identified by shape. To capture information of the image as more as possible, we can concatenate different features into a long

☆ This paper has been recommended for acceptance by M.T. Sun.
* Corresponding author.

http://dx.doi.org/10.1016/j.jvcir.2015.10.006
1047-3203/

Table 1
Keyword distribution on the Corel-5 K, ESP game and IAPRTC-12.

Models	Keywords	Images	Avg. images/keyword	Max. images/keyword	Min. images/keyword
Corel5K	260	5000	66	1120	2
ESP game	268	20,770	364	5059	20
IAPRTC-12	291	19,627	386	5534	50

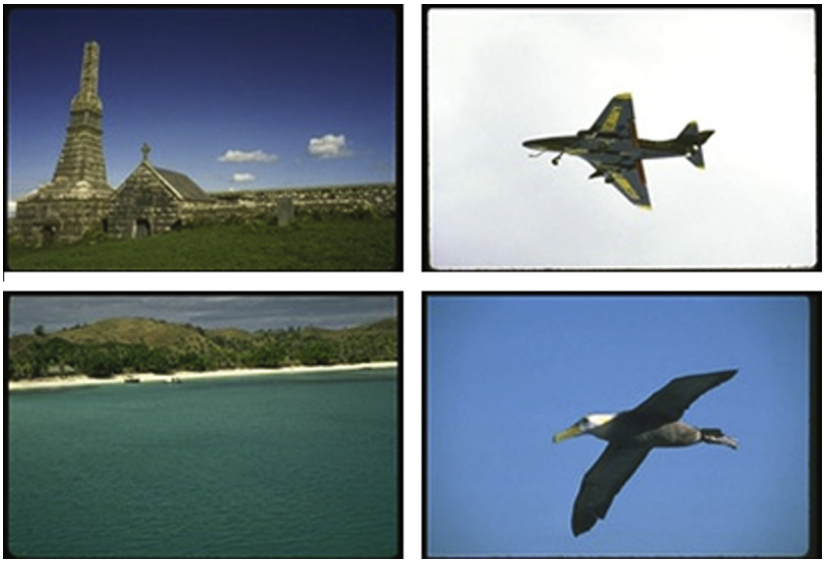

Fig. 1. Image examples that can be easily annotated by human.

vector. However, this concatenation causes overfitting in the case of a small size training sample and is not physically meaningful because each view has a specific statistical property. Therefore, how to utilize the different image feature properties is important in tagging process.

Stacked auto-encoder (SAE) [24], a typical deep learning structure, can learn extremely complicated relationships between image features and semantic information. Thus SAE is a suitable machine learning model for image understanding task. However, Compare with traditional image classification, image annotation usually has several keywords for each image. We modify the SAE model by applying sigmoid function as the SAE predictor and using ranking algorithm for generating image keywords. Further, the label number in image classification is usually limited, while the keywords for image annotation is always abundant. Due to this reason, image keywords can be seen as image text information. In this paper, we propose an iteration algorithm to retrain SAE twice.

To directly address the aforementioned limitations, we propose the multi-view stacked auto-encoder (MVSAE) with imbalanced learning. One aim of this paper is to present a simple method which can avoid low frequency keyword misclassification. Our method proposes to weigh different keywords depending on their frequency and adopt log-entropy to modify the object of SAE. For effectively utilize properties of different features, algorithm in [18] proposed a single label-specific classifier for each keyword. We tactfully incorporate the same idea into SAE and design a multi-view learning algorithm. The multi-view learning algorithm evaluates the annotation results on each keyword from different features and chooses the feature which has the best performance as the label-specific feature. As a result, each keyword can be produced by the most appropriate features. We carefully implement MVSAE for image annotation and conduct experiments on three popular image annotation data sets [14]. The experimental results demonstrate the effectiveness of MVSAE by comparing with the baseline algorithms.

The contributions of this paper are threefold:

(1) A novel SAE framework with sigmoid predictor is constructed for image annotation problem. Further, we propose a new iteration algorithm which combines the image visual features and text information as model inputs.
(2) The imbalance learning method with log-entropy weights is proposed for solving the problem of image keywords with imbalanced distribution.
(3) The multi-view SAE (MVSAE) is proposed to utilize the different descriptor properties.

The rest of this paper is organized as follows. Section 2 introduces a related work of image annotation. Section 3 describes the details of our proposed MVSAE framework. Section 4 compares our framework with the existing models and analyzes the results. Section 5 concludes the paper.

2. Related work

2.1. Advantage of deep learning

Deep neural networks, containing multiple nonlinear hidden layers, can learn very complicated relationships between their inputs and outputs. However, the nonlinear mapping between the inputs and outputs makes network be prone to local optimum and difficult to converge by back-propagation algorithm [20]. In order to overcome the learning problem of deep neural network,

Hinton et al. [21] proposed unsupervised layer-wise greedy training algorithm based on the Restricted Boltzmann Machines (RBM). In addition, deep learning methods also appeared in many deformed structures such as convolution neural network (CNN) [22], deep boltzmann machines (DBM) [23], and auto-encoder (AE) [24].

Pretraining followed by finetuning with backpropagation has been shown to give significant performance boosts over finetuning from random initializations in certain cases. Generally there are three ways for pretraining: RBM, DBM and AE. The RBM, the first model for pretraining, has been used effectively in modeling distributions over binary-valued data. Further, as an extension of RBM, the DBM is a network of symmetrically coupled stochastic binary units. The AE is modified from neural network by using inputs as targets. Among these three models, DBM has the best performance, but its learning process is complicated and time-consuming. Vincent et al. [24] presented the denoising auto-encoder (DAE), in which the AE is modified by adding noise into inputs. The DAE has comparable performance with DBM in image application, and its learning process is much simpler than DBM. Thereby we choose DAE to accomplish unsupervised pretraining in this paper.

In practice, deep learning algorithm is widely used in the object recognition, image classification, speech recognition and other fields, while it has done seldom work on image annotation task. Srivastava et al. [25] combined two modified Restricted Boltzmann Machines (RBM) (Gaussian RBM for image, Replicated Softmax model for text) with a common latent layer to construct a Multi-Modal Deep Belief Network. The model can be used to generate such missing keywords by clamping the observed image features at the image path (Guassian RBM path) and sampling the hidden modalities from the conditional distribution by running the standard alternating Gibbs sampler. Socher et al. [26] and Pinheiro et al. [27] used recurrent neural network for scene labeling by assigning a class label to each pixel in an image neither relying on any segmentation technique nor any task-specific features. Wang et al. [28] proposed an effective multi-model retrieval based on stacked auto-encoder which project high-dimensional features extracted from data of different media types into a common low-dimensional space for metric learning.

2.2. Imbalance learning

Imbalanced distribution of image keywords is widely existed in images from the Internet. For instance, "sky" and "water" appear much more frequently than "hotel" and "museum" in images. If we treat each keyword equally, the low frequency keywords would be omitted during annotation process. The reason is that model selection criteria managing the bias/variance tradeoff often are more sensitive to larger labels than to smaller labels. For the sake of learning better on all labels, cost-sensitive methods are proposed [37].

Cost-sensitive methods focus on the imbalanced learning problem by using different cost matrices that describe the costs for any misclassifying particular data example. They design optimal classifiers with respect to losses which weigh certain types of errors of training examples more heavier than others. For instance, Guillaumin et al. [14] took into account the imbalance among keywords by incorporating cost for keyword prediction. Cost-sensitivity can be also introduced to SAE by the most effective way: making the object cost-sensitive [38].

2.3. Multi-view learning

Images are often described by many views. The concept of 'views' used for images refers to different features or attributes for depicting the objects to be classified. Schemes in Multi-view learning utilize the features from different views to boost image classification performance. Extensive work has been done in the area of multi-view learning for image understanding. Images measured by different views were used to construct a prior and formulate a regularization term for semi-supervised boosting algorithm in [36]. Guillaumin et al. [14] combined different visual representations with the keyword feature for semi-supervised image classification. Liu et al. [19] presented the multi-view Hessian discriminative sparse coding which seamlessly integrates Hessian regularization with discriminative sparse coding for multi-view learning problems. Hu et al. [18] used label-specific feature learning algorithm to find a suitable feature space for each keywords. Luo et al. [35] introduced multi-view vector-valued manifold regularization to integrate multiple features for multi-label image classification.

3. The multi-view stacked auto-encoder framework

In this section, we introduce the framework of our proposed model. Fig. 2 shows the whole image annotation scenario. The core of the MVSAE model is the basic SAE which is used to build the mapping from image features to keywords. Step 1 (Feature Extraction), for effectively utilizing different image descriptors, we build the feature pool extracted from images. Step 2 (Imbalance Learning), considering the image keywords with imbalanced distribution, we introduce the imbalance learning model to weigh different keywords depending on their frequency and generate the SAE optimal object. Step 3 (Model Learning), different features are imported to SAE for learning model parameters. Step 4 (Keyword Distribution Generating), SAE model generates different tag distribution for given images according to different image views. Step 5 (Multi-view Learning), the multi-view model is applied to find the label-specific view for each keyword and generate the final annotation results. In the following, the details for each step will be presented.

3.1. Problem formulation

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$ denote the dictionary of M possible annotation keywords. We consider the typical image annotation task with N pairs as $\{(I_1, T_1), (I_2, T_2), \ldots, (I_N, T_N) \in \mathbb{R}^d \times \{0,1\}^M\}$, which we suppose to be i.i.d samples from the database. The ith image is represented by $I_i = \{I_i^1, I_i^2, \ldots, I_i^d\} \in \mathbb{R}^d$ and labeled with $T_i = \{t_i^1, t_i^2, \ldots, t_i^M\} \in \{0,1\}^M$. In image keyword T_i, $t_i^j = 1$ means that the ith image is labeled with the word ω_j, and $t_i^j = 0$ means that the image I_i is not labeled with the word ω_j. Here, we assume that the image keywords given by data sets are correct and complete. The goal of semantic image annotation is to extract the set of semantic labels for a given image.

3.2. Stacked auto-encoder

Consider solving the image annotation problem by using deep neural networks. As ordinary, image features are used as model inputs and keywords are used as model object. Meanwhile, several hidden layers are applied for modeling the complex relationship between features and tags. After the deep neural network well trained, the model could generate suitable keywords for new images. Due to the fact that the performance of the deep neural network is highly depended on initial parameters, the model optimization method is a problem. As mentioned before, the pretraining is an efficient way for overcome this trouble.

As illustrated in the blue rectangle of Fig. 3, the typical deep neural network training framework is shown. In comparison to

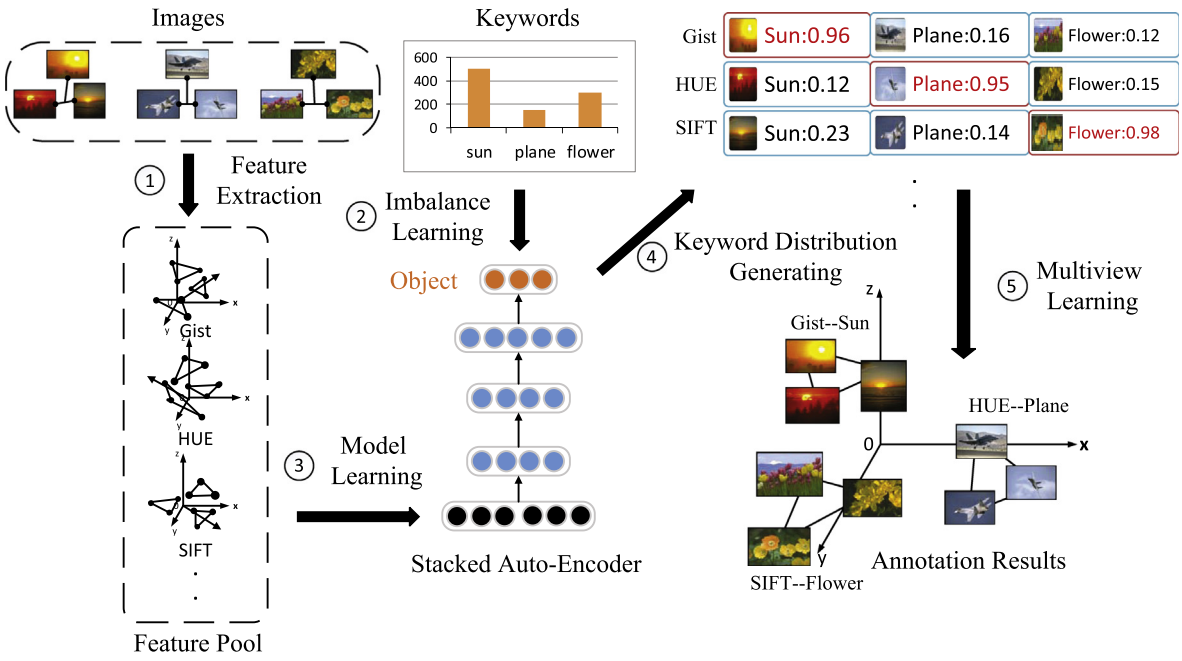

Fig. 2. Multi-view stacked auto-encoder (MVSAE) model.

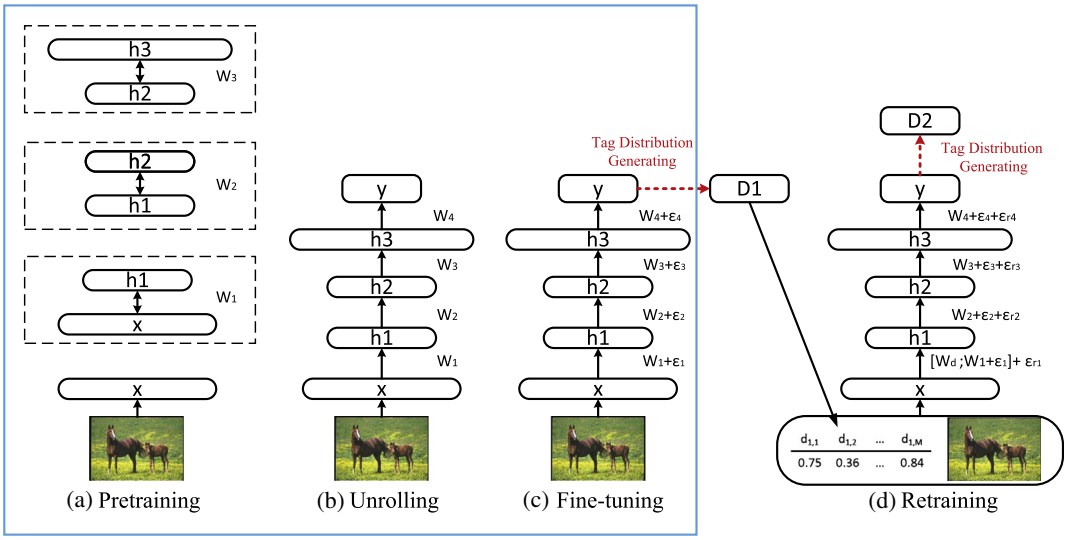

Fig. 3. Learning process for the iteration deep neural network.

traditional learning process of neural network by directly optimizing the model with random initial parameters, the deep learning can be divided into three stages: a layer-wise unsupervised pretraining, unrolling and fine-tuning. In the first stage, the image features x are used for learning the first layer parameter W_1 with unsupervised learning model AE. When the first AE is well trained, W_1 is got to generate the hidden layer h_1, which will be used as the inputs of the second AE. As the same way in learning W_1, the parameters W_2 and W_3 are learned. In the second stage, the deep neural network is initialized with the parameters W_1, W_2, W_3 learned in the first stage and W_4 generated randomly. In the last stage, the backpropagation algorithm is used to optimize the whole deep neural network as traditional neural network. The final optimal parameters can be wrote as $W_1+\epsilon_1$, $W_2+\epsilon_2$, $W_3+\epsilon_3$ and $W_4+\epsilon_4$ which mean fine-tuning on pretraining parameters W_1, W_2, W_3 and W_4. Generally, the deep neural networks with pretraining by AE are also called as Stacked Auto-Encoder (SAE). Because the deep neural network in this model is stacked with several Auto-Encoders.

AE derived from neural network is an effective algorithm for unsupervised learning as shown in Fig. 4a. Suppose we only have a set of unlabeled training examples $\{x_1, x_2, x_3, \ldots\}$, where $x_i \in \mathbb{R}^n$. The AE model can learn good representation of inputs x by setting the AE object to be equal to the inputs x. As shown in Fig. 4a, the AE is composed by two parts: encoder f_θ and decoder $g_{\theta'}$.

Encoder f_θ transforms an input vector x into hidden representation h. Its typical form is followed by a nonlinearity:

$$f_\theta(x) = \sigma(Wx + b)$$

Its parameter set is $\theta = \{W, b\}$, where W is the weight matrix and b is an bias vector. $\sigma = 1/(1+\exp(-x))$ is the activation function.

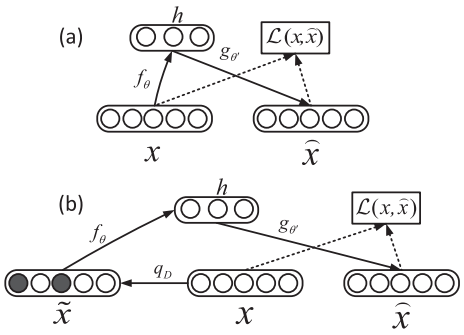

Fig. 4. (a) auto-encoder structure. (b) denoising auto-encoder structure.

Decoder $g_{\theta'}$ maps h back to a reconstructed dimensional vector $\hat{x}$ in input space. Its typical form depends on the AE inputs:

$$g_{\theta'}(h) = \begin{cases} \sigma(W'h + b') & x \in [0,1] \\ W'h + b' & x \in \Re \end{cases}$$

with appropriate parameter $\theta' = \{W', b'\}$.

AE tries to learn an approximation to the identity function so as to output $\hat{x} = g_{\theta'}(f_\theta(x))$ which is similar to x. We define the loss function with respect to our model to be $\mathcal{L}(x,\hat{x})$, then the model can be learned through minimizing the loss function as following optimization:

$$\theta^*, \theta'^* = \arg\min_{\theta,\theta'} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(x_i, g_{\theta'}(f_\theta(x_i))) \tag{1}$$

Generally, the auto-encoder is trained by back-propagation algorithm, and the loss function can be squared error loss or cross-entropy loss. In addition, the tied weights between W and W' follow a strict constraint $W' = W^T$.

Unfortunately, the reconstruction criterion alone is unable to guarantee the extraction of useful features as it can lead to the obvious solution "simply copy the input" or similarly uninteresting ones that trivially maximize mutual information [24]. Denoising auto-encoder (DAE) is proposed to change the reconstruction criterion by optimizing the reconstruction of a clean "repaired" input from a corrupted version of it as shown in Fig. 4b. Compared with AE, DAE is done by first corrupting the initial input x into $\tilde{x}$ by means of a stochastic mapping $\tilde{x} \sim q(\tilde{x}|x)$ (Masking noise: a fraction of the elements of x (chosen at random for each example) is forced to 0). Then the corrupted input $\tilde{x}$ is then mapped to a hidden representation $h = f_\theta(\tilde{x})$ from which we reconstruct a $\hat{x} = g_{\theta'}(h)$. At last, the parameters $\{\theta, \theta'\}$ are trained to minimize the average reconstruction error over a training set ($\hat{x}$ as close as possible to the uncorrupted input x). Therefore, DAE could generate a higher level stable and robust representation under corruptions of the input. The optimization objective can be written as follows:

$$\theta^*, \theta'^* = \arg\min_{\theta,\theta'} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(x_i, g_{\theta'}(f_\theta(\tilde{x}_i))) \tag{2}$$

Consider a SAE with L hidden layers for image annotation. Let $l \in \{1,\ldots,L\}$ index the hidden layers of the SAE. Let h^l denote the vector of outputs from layer l ($h^0 = I$ is the input and h^{L+1} is the output). W^l and b^l are the weights and biases at layer l. As mentioned before, $\{W^l, b^l\},\ l \in \{1,\ldots,L\}$ are initialized with the results of layer-wise pre-training by using DAE, and $\{W^{L+1}, b^{L+1}\}$ are initialized randomly. The feed-forward operation of a standard neural network can be described as: for $l \in \{0,\ldots,L-1\}$,

$$h^{l+1} = \sigma(W^{l+1}h^l + b^{l+1})$$
$$h^{L+1} = P(W^{L+1}h^L + b^{L+1})$$

where σ is any activation function and P is the prediction function. There are two common prediction function including softmax function ($a_j = e^{a_j}/\sum_k e^{a_k}$, where a_j means the jth) output neuron) and sigmoid function ($a_j = 1/(1+e^{-a_j})$, where a_j means the jth output neuron) for SAE. Softmax function grades image candidate classes correlatively and chooses the biggest one as the image label. It is suitable for image classification with classes mutually excluded (for example, images can be classified into "car" or "airplane" solely). However, keywords in image annotation are usually mutually related (for example, images can be annotated with "car" and "road" meanwhile). In comparison to softmax function, sigmoid function has the advantages of grading each image candidate keywords independently and returns the top ranked keywords as the annotations for images. So Sigmoid function is chosen as the prediction function in this paper. The whole model is trained by using back-propagation algorithm for solving the optimization problem in (3).

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}(F_\theta(I_i), T_i) \tag{3}$$

where $F(I) = P_{\theta_{L+1}}(\sigma_{\theta_L}(\ldots(\sigma_{\theta_1}(I))))$ is the compound function of the SAE, and θ are model parameters $\{W^l, b^l\},\ l \in \{1,\ldots,L+1\}$. Cross-entropy loss ($\mathcal{L}(X,Y) = X\log(Y) + (1-X)\log(1-Y)$) is defined to be the loss function in our model for its efficiency and simplicity. This loss function has been widely adopted in practice, when the objective element $T \in [0,1]$. After the model well trained, the SAE output layer h^{L+1} can be seen as the image keyword probability distribution D for given images. Finally, the image keywords $\widehat{T}$ are obtained by ranking D.

In image annotation problem, images always have abundant keywords, and visual features and text information can be both seen as image representation. Multi-modal classifier unites the image visual features and texts together for image classification, which outperforms the model using visual features only. In this paper, we develop a SAE iterative algorithm shown in Fig. 3 by retraining SAE twice. At First, visual feature I as model inputs x is used to train the SAE for generating the initial keyword probability distribution D^1 which denoted by red dash arrow. Then I and D^1 together compose new representation of images as model inputs x, which is used to retrain the SAE model for generating the final keyword probability distribution D^2. Finally, image keywords $\widehat{T}$ are obtained from D^2. During the second training step, the parameters of the first layer is augmented to $[W_d;\ W_1 + \epsilon_1]$ in which W_d is initialized randomly and $W_1 + \epsilon_1$ follows the result in the first step. As illustrated in Fig. 3, model parameters after retraining can be written as $[W_d;\ W_1 + \epsilon_1] + \epsilon_{r1},\ W_2 + \epsilon_2 + \epsilon_{r2},\ W_3 + \epsilon_3 + \epsilon_{r3}$ and $W_4 + \epsilon_4 + \epsilon_{r4}$. Unlike the text information being the number of keywords in Multi-modal classifier, the D^1 is probability distribution having the similar structure with visual features. Consequently, SAE treats D^1 as the image additional feature and only owns one path for efficiency.

3.3. Imbalance learning for keywords with imbalanced distribution

Imbalanced distribution of image keywords make classifiers tend to provide a severely skew degree of accuracies. The majority keywords have high percent accuracies, while the minority keywords have low percent accuracies. For addressing this problem, the object of the SAE is altered to bias the model to focus more on keywords with low frequency. Inspired by the study in [30], we modify the object of SAE by incorporating the log-entropy

weighing scheme based on information theory. Assuming that there are N images and M different keywords in the data set. The occurrence amount of jth keyword in training set is denoted as $S^j = \sum_{i=1}^{N} t_i^j$, where t_i is the ith image keywords. Due to the fact that each keyword is used to annotate at least one image, thus $S^j > 0$. Accordingly, the weight for jth keyword in ith image π_i^j can be calculated as follows:

$$\pi_i^j = \left(1 + \sum_{k=1}^{N} \frac{p_k^j \log p_k^j}{\log N}\right) \cdot \log(t_i^j + 1) \tag{4}$$

where $p_k^j = t_k^j / S^j$. Let $\Pi = [\pi_{i,j}]_{N \times M}$, with each of its row L2-normalized by one, include the keyword weights assignments for all the training images. Further, the optimum function of SAE can be expressed as follows:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}(F_\theta(I_i) - \Pi_i * T_i) \tag{5}$$

where Π_i denotes the ith row of Π, and $*$ denotes element-wise dot.

3.4. Multi-view learning for label-specific features

Numerous features have been proposed for compute vision problem, and the use of proper features determines the performance of learning algorithms. For instance, Unfortunately no features can solve the semantic-gap problem on all keywords efficiently. Inspired by [18,19], we adopt multi-view method based on the assumption that high-level semantic concepts can be reflected in a suitable low-level feature for image annotation. We can understand this assumption in this way. Some semantic concepts can be predicted quite well using color information only, while the prediction of some other concepts may need shape information. If human knowledge can be employed in the annotation system, the model can choose more suitable features for a test image. Therefore, in the rest of this section, we will present the multi-view learning algorithm for training an accurate and efficient image annotation model.

Now, we present the learning process in multi-view learning algorithm. Given a data set including N images, we extract different visual features and build a feature pool first. Using the single feature or the combination of several features, we get V different views $\{I_{1,v}, I_{2,v}, \ldots, I_{N,v}\}_{v=1}^{V}$ for each image. Therefore, for the ith image, we achieve V different image annotation results from V different views. The annotation results are denoted by the matrix R_i, where the vth row elements $R_{i,v}$ represent the vth view annotation result. For comprehensive analysis, the criterion denoted by $F1$ (details in next section) are chosen to evaluate the annotation results. The goal of finding the suitable view for a specific keyword is to select the view which has the highest $F1$ score to generate such keyword. We define the vector c to record the best view for all keywords, and each element $c(j),\ j = 1, 2, \ldots, M$ can be calculated as follows:

$$c(j) = \arg\max_{1 \leqslant v \leqslant V} F1_{vj},\ j = 1, 2, \ldots, M \tag{6}$$

Consequently, the $c(j)$th view is label-specific view for the jth keyword. To estimate the performance of our model, we take cross-validation to get the appropriate vector c. Based on vector c, a M rows and V columns factor matrix F can be obtained, wherein each row is zero except the position of $\{j, c(j)\}$ is equal to one. Finally, the right keywords can be calculated by:

$$\widehat{T}_i = \sum_{v=1}^{V} F * R_i,\ i = 1, 2, \ldots N \tag{7}$$

where $*$ denotes element-wise dot.

In this paper 9 different visual descriptors [14] are extracted for image annotation. These features include a GIST feature, 2 Hue features, 2 SIFT features, 2 Harris + Hue features, and 2 Harris + SIFT features. Then the quantized descriptors are represented by a visual word histogram (e.g. "Dense Hue" and "Harris SIFT"). Subsequently a new histogram representation which encodes spatial information on each histogram is constructed by computing over a 3×1 horizontal decomposition of the image (e.g. "Dense HueV3H1" and "Harris SIFTV3H1"). All these features in single forms or combing forms are feed into feature pool to constitute the different view $I_{i,v}$ for image I_i.

3.5. MVSAE algorithm

Algorithm 1 lists the detailed procedure of MVSAE. The algorithm consists of three major components: the pro-processing stage (line 1–2) which builds different image views and modifies SAE object, iterative training stage (line 3–7) which combines keyword probability distribution got by first iteration and image features for image annotation, and multi-view method stage (line 8–14) which searches for the label-specific view to generate final annotation results. In the algorithm, we use subscript *Tr* to indicate the data from training set, subscript *Va* to indicate the data from validation set, and subscript *Te* to indicate the data from test set. $[I;\ D^1]$ denotes combining I and D^1 together as model inputs.

Algorithm 1. Image Annotation Based on Multi-view Stacked Auto-encoder

Input: Training set I_{Tr}, T_{Tr}, Validation set I_{Va}, T_{Va}, Test set I_{Te}
Output: Test image keywords T_{Te}

1 Compute modified object $\widetilde{T}_{Tr} = \Pi * T_{Tr}$ according to Eq. (4)
2 Build image views $I_{Tr,v}$, $I_{Tr,v}$, $I_{Tr,v}$, $v = \{1, 2, \ldots, V\}$ from I_{Tr}, I_{Va}, I_{Te}
3 **for** $v = 1$ **to** V **do**
4 Train model by using $I_{Tr,v}$, $\widetilde{T}_{Tr}$
5 Get initial keyword distribution probabilities $D^1_{Tr,v}$, $D^1_{Va,v}$ and $D^1_{Te,v}$ from $I_{Tr,v}$, $I_{Va,v}$ and $I_{Te,v}$
6 Train model by using $[I_{Tr,v};\ D^1_{Tr,v}]$, $\widetilde{T}_{Tr}$
7 Get final keyword distribution probabilities $D^2_{Tr,v}$, $D^2_{Va,v}$ and $D^2_{Te,v}$ from $[I_{Tr,v};\ D^1_{Tr,v}]$, $[I_{Va,v};\ D^1_{Va,v}]$ and $[I_{Te,v};\ D^1_{Te,v}]$
8 Rank $D^2_{Va,v}$, $D^2_{Te,v}$ and get image keywords $\widehat{T}_{Va,v}$, $\widehat{T}_{U,v}$ for each view
9 Compute $F1$ score $F1_{v,j}$, $j = \{1, 2, \ldots, M\}$ from $\widehat{T}_{Va,v}$
10 **for** $j = 1$ **to** M **do**
11 Compute $c(j)$ according to Eq. (6) for each keyword
12 Transform vector c to matrix F.
13 Compute $\widehat{T}_{Te}$ from $\widehat{T}_{Te,v}$, $v = \{1, 2, \ldots, V\}$ according to Eq. (7)
14 Return $\widehat{T}_{Te}$

4. Experimental result

In this section, we evaluate the quality of the multi-view SAE model on three standard benchmark data sets. First, we introduce the details of data sets, image features and evaluation metric. Then, we test the effectiveness of the SAE model with imbalance learning and show signification of multi-view method. Finally, we compare the performance of our model with several state-of-the-art annotation methods and give some practical examples of the image annotation.

4.1. Experimental setup

We begin with a detailed description of the data sets, image features and evaluation metric.

4.1.1. Date sets

We use the three current image annotation data sets: Corel-5 K, ESP game and IAPRTC-12, to verify our algorithm. The details are introduced as follows:

Corel-5 K. The data set contains 5000 images collected from the larger Corel CD set. Each image is manually annotated with keywords from a dictionary of 260 distinct terms. On average, each image was annotated with 3.5 keywords. The training set contains 4000 images, validation set contains 500 images and test set contains 499 images.

ESP game. The data set consists of 20,770 images of a wide variety, such as logos, drawings, and personal photos, collected for the ESP collaborative image labeling task. The images are annotated with a total of 268 keywords. Each image is associated with a maximum of 15 keywords and 4.6 keywords on average. The training set contains 15,689 images, validation set contains 3000 images and test set contains 2081 images.

IAPRTC-12. The data set consists of 19,627 images of sports, actions, people, animals, cities, landscapes and many other aspects of contemporary life. Keywords are extracted from the free-owing text captions accompanying each image. Overall, 291 keywords are used. The training set contains 15,665 images, validation set contains 2000 images and test set contains 1962 images.

For all these data sets, the training/validation/test split follows previous work.

4.1.2. Evaluation metric

For full comparability, we adopt the same evaluation metric as in [14]. First, all images are annotated with the five most relevant keywords (i.e. we rank the keywords for the test images in the descending order based on their probabilities of the output and return the top ranked keywords as the annotations for images.). Second, precision (P) and recall (R) are computed for each keyword. The reported measurements are averaged across all keywords. For easier comparability, both factors are combined in the $F1$-score ($F1 = 2P * R/(P+R)$), which is reported separately. We also report the number of keywords with non-zero recall value ($N+$). In all metrics, a higher value indicates better performance.

4.2. Effect of modified SAE

The experiments described here evaluate the effect of the modified SAE. For clearness, we define the SAE with softmax predictor as SAE-soft, the SAE with sigmoid predictor as SAE-sigm and the SAE with iteration algorithm as SAE-iter. Particularly, SAE-iter uses sigmoid function as its predictor. The same idea of combining image visual features and text information is also used in the multi-auto-encoder (Multi-AE) [28]. The Multi-AE derives from multi-model algorithm with SAE [29] which is the first multi-model deep learning algorithm combining the Audio and Video information. It builds the bilateral relationship between image and text for image annotation and retrieval. We adopt combinative features, including all dense features (SIFT, SIFTV3H1, Hue and HueV3H1) and GIST feature, to represent the image visual content.

Table 2 summarizes the average precision/recall/$F1$/$N+$ of all keywords for the different SAE models. It can be observed that SAE-sigm has much better performance than SAE-soft. It indicates that, In comparison to softmax function, sigmoid function is suitable for multi-label image annotation problem. Further, SAE-iter model slightly outperforms the SAE-sigm model, which suggests the effect of combined inputs. At last, due to the fact that Multi-AE is deemed not be a specialized model for image annotation, Multi-AE has the worst result during all these SAE models. In the following paper, SAE-iter is applied for image annotation and abbreviated to SAE for short.

Table 2
Mean annotation results of all keywords from different SAE models on the Corel-5 K. (Bold numbers report when a model outperforms all others.)

Models	P	R	$F1$	$N+$
SAE-soft	0.20	0.28	0.24	120
SAE-sigm	0.27	0.36	0.32	149
SAE-iter	**0.30**	**0.38**	**0.34**	**155**
Multi-AE	0.15	0.20	0.17	110

4.3. Advantage of imbalance learning

The experiments described here evaluate the performance of imbalance learning SAE (SAE-im) on typical different frequency keywords. In original SAE model, the object is represented by $T_i = \{t_i^1, t_i^2, \ldots, t_i^m\} \in \{0,1\}^m$, where $t_i^j = 1$ if the jth keyword appears in the ith image and $t_i^j = 0$ otherwise. In SAE-im, the object is represented by $\Pi_i * T_i$ ($*$ denotes element-wise dot) according to Eq. (4). The same features are used as before.

Table 3 gives the annotation results of several typical keywords and Mean annotation results from SAE and SAE-im on the Corel-5 K. As shown in Table 3, the imbalance learning method has different influence on different frequency level keywords. Towards high frequency keyword, $F1$ score slightly decreases, because tendency to low frequency may cause high frequency keywords misclassified. On the other hand, low level frequency keywords have better performance than original SAE. Consequently, imbalance learning can improve the performance of low frequency keywords with slightly sacrificing high frequency keywords. The last row in Table 3 shows that the performance of SAE-im outperform the SAE model on the mean annotation results of all keywords, which suggests the advantage of imbalance learning method on whole annotation performance.

4.4. Analysis of multi-view learning

In order to confirm the effectiveness of multi-view learning, the SAE-im is applied to a feature pool with single features and two mixture features: dense mix features (including Dense Sift, DenseSiftV3h1, Dense Hue, and GIST), and Harris mix features (including

Table 3
Annotation results of several typical keywords and the average of all keywords from SAE and SAE-im on the Corel-5 K. (Bold numbers report when a model outperforms all others.)

Word	Num.	SAE			SAE-im		
		P	R	$F1$	P	R	$F1$
Water	1004	0.51	0.79	**0.62**	0.53	0.77	0.62
Tree	854	0.42	0.68	**0.52**	0.37	0.61	0.46
Buildings	408	0.47	0.65	**0.54**	0.38	0.70	0.49
Snow	267	0.58	0.71	**0.64**	0.53	0.74	0.62
Stone	212	0.70	0.70	**0.70**	0.61	0.70	0.65
Sand	184	0.40	0.63	**0.49**	0.32	0.58	0.42
Cars	134	0.50	0.59	0.54	0.73	0.65	**0.69**
House	124	0.32	0.37	0.34	0.40	0.53	**0.45**
Tracks	103	0.64	0.82	0.72	0.82	0.82	**0.82**
Coral	89	0.58	0.78	0.67	0.73	0.89	**0.80**
Sunset	76	0.22	0.57	0.32	0.40	0.86	**0.55**
Petals	59	0.23	0.75	0.35	0.25	1.00	**0.40**
Average		0.30	0.39	0.34	**0.32**	**0.40**	**0.36**

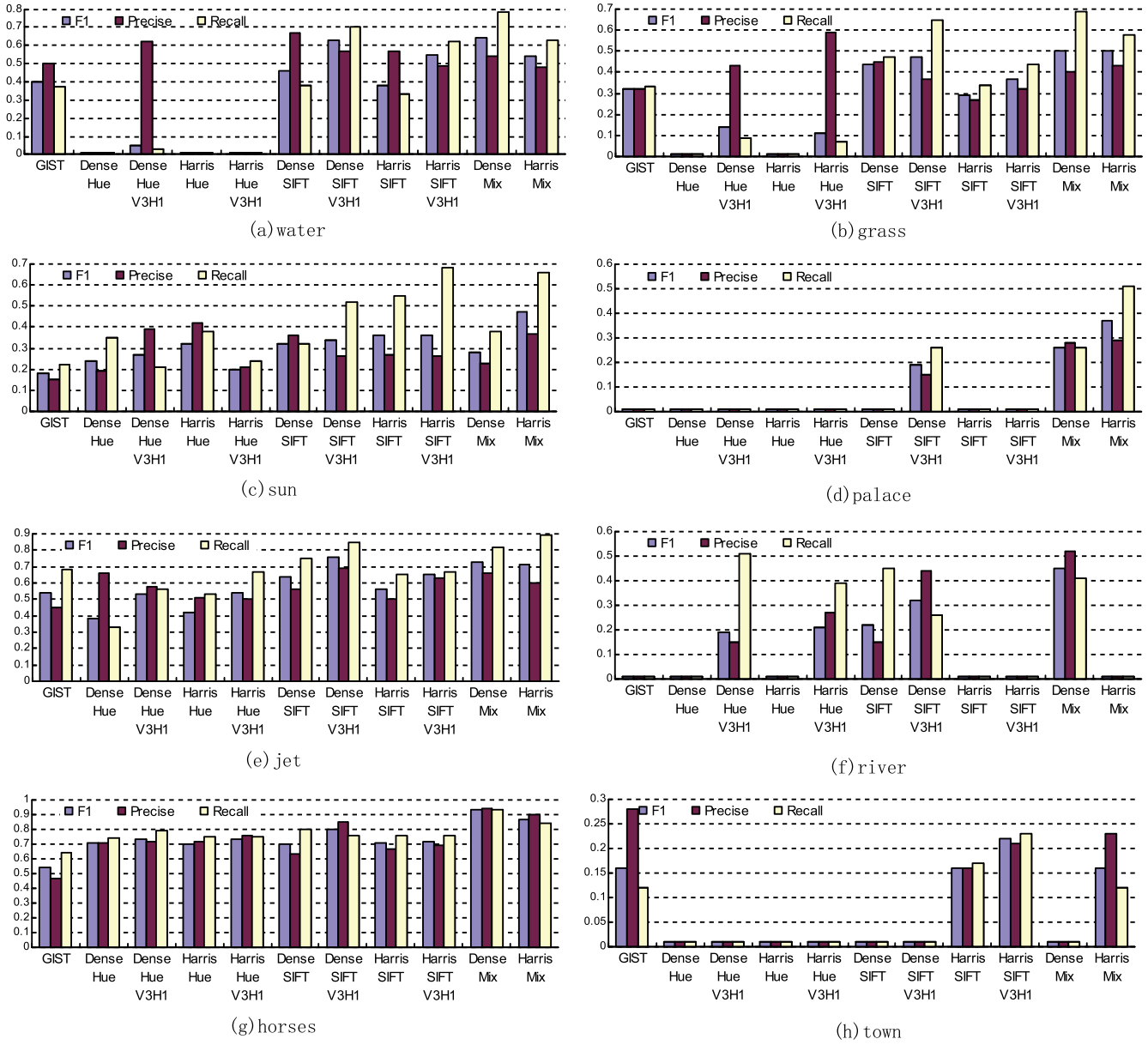

Fig. 5. The annotation results of different views on selected keywords.

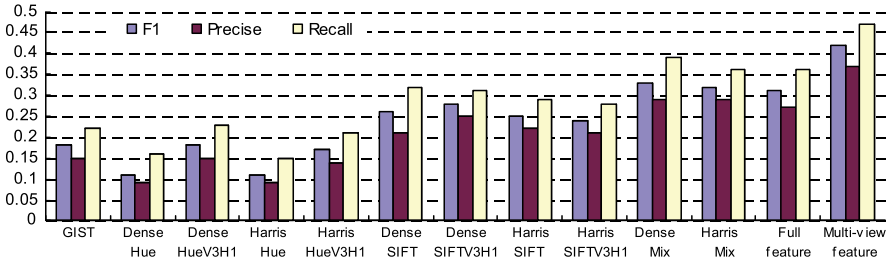

Fig. 6. The mean annotation results of different views.

Harris SIFT, Harris SIFTV3H1, Harris Hue, and GIST). Fig. 5 shows details of annotation performance of different views over selected keywords. It can be observed that, for each keyword different views achieve different performance, and some views have poor performance on several special keywords(like "grass", "palace", and "river"). Obviously, no feature always gets best annotation performance on any keyword. Therefore, it is essential to find the label-specific view for each keyword.

For utilizing different feature properties, we apply the MVSAE to find the label-specific feature (multi-view feature) for each keyword. A similar effect could have been achieved by concatenating different features into a long vector (full feature) and simply using

Table 4
Comparison of MVSAE and other baseline algorithms in terms of *P*, *R*, *F*1 score and *N*+ on the Corel-5 K, ESP game and IAPRTC-12. (Bold numbers report when a model outperforms all others.)

Models	Corel-5 K				ESP game				IAPRTC-12			
	P	*R*	*F*1	*N*+	*P*	*R*	*F*1	*N*+	*P*	*R*	*F*1	*N*+
CRM[4]	0.16	0.19	0.17	107	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SML[11]	0.23	0.29	0.26	137	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
GS[31]	0.30	0.33	0.32	146	N/A	N/A	N/A	N/A	0.32	0.29	0.32	252
MBRM[5]	0.24	0.25	0.24	122	0.18	0.19	0.18	209	0.24	0.23	0.23	223
JEC[9]	0.27	0.32	0.29	139	0.24	0.19	0.21	222	0.29	0.19	0.23	211
TagProp[14]	0.33	0.42	0.37	160	0.39	0.27	0.32	238	0.45	0.34	0.39	260
LM3L[32]	0.33	0.37	0.35	146	0.40	0.26	0.32	239	0.44	0.28	0.34	242
KSVM-VT[33]	0.32	0.42	0.37	179	0.33	**0.32**	0.33	259	0.47	0.29	0.36	268
2PKNN[34]	**0.44**	0.46	**0.45**	**191**	**0.53**	0.27	**0.36**	**259**	**0.54**	0.37	**0.44**	278
MVSAE	0.37	**0.47**	0.42	175	0.47	0.28	0.34	246	0.43	**0.38**	0.40	**283**

Table 5
Comparison of MVSAE and 2PKNN in terms of approximate testing time of 500 images in second.

Models	Corel-5 K	ESP game	IAPRTC-12
MVSAE	0.13	0.20	0.19
2PKNN	22	139	133

PCA for its dimensionality reduction. Fig. 6 shows the mean annotation results of different views. It is noteworthy that multi-view feature outperforms others, and the performance of full feature is even inferior to the dense mixture feature. It demonstrates that multi-view method can more effectively utilize the complementary among different features than full feature and significantly improve the annotation results.

4.5. Comparison to state-of-the-art

In order to verify the effectiveness of MVSAE, the MVSAE algorithm is compared with nine state-of-the-art algorithms [4,5,11,9,14,31–34] on Corel-5 K, ESP game and IAPRTC-12. Table 4 shows detailed comparisons between MVSAE algorithm and other algorithms. It is noteworthy that our recall score get best performance on Corel-5 K and IAPRTC-12 due to enhancing the low frequency keywords by imbalance learning. For the synthetical score *F*1, our model significantly outperforms the other methods and aligns with the state-of-the-art Algorithm PKNN. Moreover the present data confirms the role of multi-view learning. However, as we demonstrate next, MVSAE achieves enormous speedup over 2PKNN in testing process.

Although 2PKNN achieves superior performance on several benchmark datasets, $O(n)$ test complexity hinders its applicability to large scale datasets (where n is the number of training samples). Table 5 shows the approximate testing time of 500 new images required by MVSAE and 2PKNN on these three datasets (All experiments were conducted on a desktop with dual 2-core Intel i5 cpus with 3.2 GHz). Obviously, the testing time consumed by MVSAE model is much shorter than the time consumed by 2PKNN model, and grows little with the increasing training sample number. So the MVSAE model is suitable for large scale image annotation task.

Fig. 7 presents some examples of image annotation produced by MVSAE algorithm. Unlike traditional methods, which always provide images with the same number of keywords (usually 5), our model can annotate images with different number of keywords according to their features. It is more reasonable for images in real situation. Here we give two annotation levels to analyze the real image annotation effectiveness. Completely correct tagged images are listed in the first row. In this level, the model can annotates the

Fig. 7. Keywords predicted by using MVSAE for several images on the Corel-5 K.

images with most related keywords besides the ground truth. The second row is the images annotated with partial correct keywords. In this level, some keywords are missing for their abstractive concepts and keywords with similar image texture are easily confusing.

5. Conclusion

In this paper, we have presented a novel MVSAE model to jointly build the correlations between low-level image features and high-level semantic keywords to realize automatically images annotation. First, we modify the SAE by using sigmoid function predictor and iteration algorithm. Second, for solving the image keywords with imbalanced distribution, we apply the imbalance learning method to weigh different frequency keywords. Third, we introduce the multi-view method to the model for the complementary information between different features can boost the tagging performance. The proposed algorithm is verified in three typical data sets, corel-5 K, ESP game and IAPRTC-12. The experiment results show that the multi-view stacked auto-encoder with imbalance learning can effectively realize the image annotation automatically, and get annotation results as well as the best techniques in the baseline.

The multi-view stacked auto-encoder can be easily incorporated into an image tagging tool, where for large amount of given images the well-trained model will perform well and automatically recommend keywords for the images. Building such a tool will make image annotation more efficient, less labor intensive, and ultimately help retrieve the huge number of images on the Internet more easily.

References

[1] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: State of the art and challenges, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 2 (1) (2006) 1–19.

[2] L. Wenyin, S Dumais, Y Sun, H. Zhang, M. Czerwinski and B. Field, Semi-automatic image annotation, in: Proc. of interact: conference on HCI. 2001, pp. 326-333.

[3] J. Jeon, V. Lavrenko, R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval(ACM), 2003, pp.119-126.

[4] V. Lavrenko, R. Manmatha, J. Jeon, A model for learning the semantics of pictures, in: Advances in neural information processing systems, 2003.

[5] S.L. Feng, R. Manmatha, V. Lavrenko, Multiple bernoulli relevance models for image and video annotation, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR), 2004, pp.1002-1009.

[6] K. Barnard, P. Duygulu, D. Forsyth, F. Nando, B. David, M.I. Jordan, Matching words and pictures, The Journal of Machine Learning Research 3 (2) (2003) 1107–1135.

[7] D.M. Blei, M.I. Jordan, Modeling annotated data, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (ACM), 2003, pp. 127-134.

[8] S.C.H. Hoi, W. Liu, M.R. Lyu, W. Ma, Learning distance metrics with contextual constraints for image retrieval, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 2072-2078.

[9] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: Proceeding of the 10th European Conference on Computer (ECCV), 2008, pp. 316-329.

[10] J. Li, J.Z. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (9) (2003) 1075–1088.

[11] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 394–410.

[12] C. Cusano, G. Ciocca, R. Schettini. Image annotation using SVM, in: International Society for Optics and Photonics on Electronic Imaging, 2003, pp. 330-338.

[13] D. Grangier, S. Bengio, A discriminative kernel-based approach to rank images from text queries, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (8) (2008) 1371–1384.

[14] M. Guillaumin, T. Mensink, J. Verbeek J, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: IEEE 12th International Conference on Computer Vision (ICCV), 2009 pp. 309-316.

[15] N. Zhou, W.K. Cheung, G. Qiu, X. Xue, A hybrid probabilistic model for unified collaborative and content-based image tagging, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1281–1294.

[16] M. Chen, A. Zheng, K. Weinberger. Fast image tagging, in: Proceedings of The 30th International Conference on Machine Learning (ICML), 2013, pp. 1274-1282.

[17] L. Wu, R. Jin, A.K. Jain, Tag completion for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (3) (2013) 716–727.

[18] J. Hu, K.M. Lam, An efficient two-stage framework for image annotation, Pattern Recognition 46 (3) (2013) 936–947.

[19] W. Liu, D. Tao, J. Cheng, Y. Tang, Multiview hessian discriminative sparse coding for image annotation, Computer Vision and Image Understanding 118 (1) (2014) 50–60.

[20] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Why does unsupervised pre-training help deep learning, The Journal of Machine Learning Research 11 (2010) 625–660.

[21] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11)(1998) 2278-2324.

[23] R. Salakhutdinov, G.E. Hinton, Deep boltzmann machines, in: International Conference on Artificial Intelligence and Statistics. 2009, pp. 448-455.

[24] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, The Journal of Machine Learning Research 9999 (2010) 3371–3408.

[25] N. Srivastava, R. Salakhutdinov. Learning representations for multimodal data with deep belief nets, in: International Conference on Machine Learning Workshop, 2012.

[26] R. Socher, C.C. Lin, C. Manning, A.Y. Ng, Parsing natural scenes and natural language with recursive neural networks, in: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011, pp. 129-136.

[27] P. Pinheiro, R. Collobert. Recurrent convolutional neural networks for scene labeling, in: Proceedings of The 31st International Conference on Machine Learning (ICML). 2014, pp. 82-90.

[28] W. Wang, B.C. Ooi, X. Yang, D. Zhang, Y. Zhuang, Effective Multi-Modal Retrieval based on Stacked Auto-Encoders, in: Proceedings of the PVLDB, 2014, pp. 649-660.

[29] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011, pp. 689-696.

[30] Z. Feng, R. Jin, A. Jain, Large-scale Image Annotation by Efficient and Robust Kernel Metric Learning, in: 2013 IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1609-1616.

[31] S. Zhang, J. Huang, Y. Huang Y, Yu, H. Li, Automatic image annotation using group sparsity, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3312-3319.

[32] B. Hariharan, L. Zelnik-Manor, M. Varma, S.V.N. Vishwanathan, Large scale max-margin multi-label classification with priors, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 423-430.

[33] Y. Verma, C.V. Jawahar, Exploring svm for image annotation in presence of confusing labels, in: Proceedings of the 24th British Machine Vision Conference, 2013.

[34] Y. Verma, C.V. Jawahar, Image annotation using metric learning in semantic neighbourhoods, in: Proceeding of the 12th European Conference on Computer (ECCV), 2012, pp. 836-849.

[35] Y. Luo, D. Tao, C. Xu, et al., Multiview vector-valued manifold regularization for multilabel image classification[J], IEEE transactions on neural networks and learning systems 24 (5) (2013) 709–722.

[36] A. Saffari, C. Leistner, M. Godec, et al., Robust multi-view boosting with priors, in: Computer VisionCECCV 2010, Springer, Berlin, Heidelberg, 2010, pp. 776–789.

[37] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[38] M.Z. Kukar and I. Kononenko, Cost-Sensitive Learning with Neural Networks, in: European Conf. Artificial Intelligence, 1998, pp. 445-449.