LEARNING OBJECT CLASSES FROM IMAGE THUMBNAILS THROUGH DEEP NEURAL NETWORKS

Erkang CHEN[1], *Xiaokang YANG*[1,3], *Hongyuan ZHA*[2], *Rui ZHANG*[1], *Wenjun ZHANG*[1]

[1]Institute of Image Communication and Information Processing
Shanghai Jiao Tong University, China
[2]College of Computing, Georgia Institute of Technology, USA
3 Institute for Computer Science, University of Freiburg, Germany

ABSTRACT

We propose a new approach for recognizing object classes which is based on the intuitive idea that human beings are able to perform the task well given only thumbnails (coarse scale version) of images. Unlike previous work which uses local image features at fine scales, our approach uses thumbnails directly, and captures their high-order correlations at coarse scales through deep multi-layer neural networks based on Restricted Boltzmann Machines. Specifically, the pretraining stage of such networks takes on the role of feature extraction. Experimental results show that the proposed approach is comparable to other state-of-the-art recognition methods in terms of accuracy. The merits of the proposed approach come from the simplicity of the workflow and the parallelizability of the implementation structure.

Index Terms— Object Class Recognition, Thumbnail, Deep Neural Networks, High-order Correlations

1. INTRODUCTION

Recognizing generic classes of objects is a challenging task in computer vision, and approaches based on local features have drawn much attention in recent years. These local features capture preliminary appearance structures, such as corners, edges and blobs. In the bag-of-words models[1, 2], usually several hundreds of local patches are extracted, processed and vector quantized for each image, forming a robust representation of the image. However, such process may take considerable time. Part based models[3, 4] discover sparse parts with strong, category-discriminative power, and learn the spatial relations of these parts, but tend to be complicated to some extent.

In this paper, we follow the intuitive idea that even when an image is scaled down to a thumbnail (i.e., coarse scale version), human vision is still able to recognize easily which classes of object they contain. One possible explanation is that, while the real world often features high-order correlations that must be included if our description about it is to be effective[5], for the task of object class recognition, high-order correlations at a coarse scale are sufficient for human vision to make correct decisions. Thus, we believe that the thumbnails of images contain sufficient information about object classes.

In order to learn strong, high-order correlations of these thumbnails and perform categorization, we adopt deep multi-layer neural networks. In particular, we use the greedy layer-wise method to speed up the training of such deep networks [6]. Experiments show that our approach achieve comparable accuracy to other state-of-the-art methods. Moreover, apart from down sampling images to obtain thumbnails and passing them through the neural networks, our method does not require explicit feature extraction. The proposed scheme is suitable for real-time applications, and is more amendable to hardware implementation, as its workflow meets KISS (Keep It Simple, Stupid) principle and its associated network is structured with high-degree of parallelism [6].

The rest of this paper is organized as follows. Section 2 discusses object class recognition, and thumbnail formation. Section 3 describes how to train deep networks in order to capture high-order correlations, followed by experimental results in section 4. Finally, we conclude our paper with some discussions.

2. THUMBNAILS FOR OBJECT CLASS RECOGNITION

While it is easy for human beings to recognize thousands of object classes, it is hard for machines to create categories, summarize common features and perform recognition. One of the key issues is the presence of large intra-class variations.

Bag-of-words models present a robust way of image representation. Usually, hundreds of local patches are extracted from each image, as illustrated in Fig. 1(b). These patches need to be processed and vector quantized to a dictionary

This work was supported in part by NSFC (60502034, 60625103), Hi-Tech Research and Development Program of China 863 (2006AA01Z124), NCET-06-0409, the 111 Project, and Alexander von Humboldt Foundation.

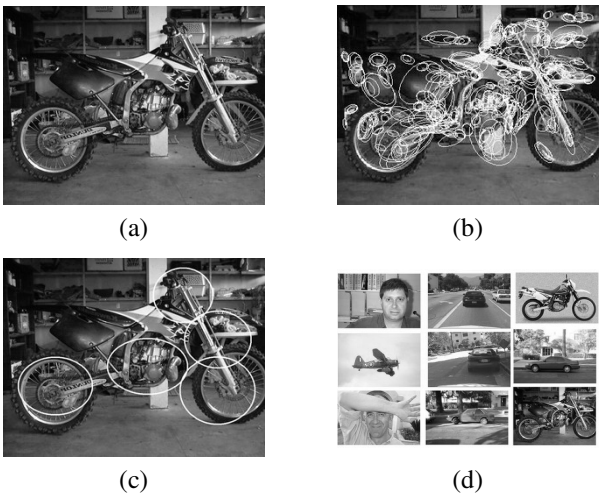

Fig. 1. (a) A motorbike image. (b) Local features in bag-of-words models. (c) Local features in part based models. (d) Thumbnails.

formed during the training stage. Then images are considered as visual documents, each represented as a histogram of visual words from the dictionary, ignoring their spatial relations. There are many local feature detectors, most of which operate in scale-space[7]. Part based models discover sparse parts that are strongly correlated with object classes (Fig. 1(c)), and learn their spatial relations. However, one problem facing such models is the correspondence between the model and the image.

From the viewpoint of high-order correlations, bag-of-words models capture local high-order correlations at fine scale, discarding global high-order correlations. Part based models take both local and global high-order correlations at fine scale into account. Our proposed approach differs from these methods, in that high-order correlations at coarse scale, i.e., thumbnails, are used to recognize object classes. We also note that high-order correlations may be quite different at different scales.

Our method does not require explicit extraction of features. Instead, raw pixel values of thumbnails are used directly. However, during the pretraining stage of the deep networks as described in section 3.1, each hidden layer can be seen as a combination of implicit feature extractors.

In the Internet, as the web image search engines such as Google Image usually present the at-first-glance search results with the form of thumbnails, we can directly learn the object classes of thumbnails thus avoiding the need to access to the original images in the hyperlinks. In local image archives, after the original images are down sampled to generate thumbnails, the dimensionality of data is greaty reduced. Given an original image of size 800×600, if we use a thumbnail of size 40×30, a ratio of 400 for dimensionality reduction is achieved, and the deep networks can be trained on the final 1200-dimensional data efficiently.

3. HIGH-ORDER CORRELATIONS LEARNING

To recognize object classes in thumbnails, we need to obtain high-order correlations at coarse scales with "deep architectures" that can represent high-level abstractions. In order to capture these statistics, we propose to train deep networks with thumbnails and their class labels . Such networks have multiple hidden layers (Fig.2(b)), which can be seen as progressive feature detectors. Lower layers extract low-level features of the data, while upper layers represent more "abstract" concepts[8], and capture strong high-order correlation of data. In our classification scenario, softmax output units are connected to the top layer, and multi-class cross-entropy error are used in backpropagation through the networks. The whole training procedure consists of two stages, pretraining and fine-tuning described as follows.

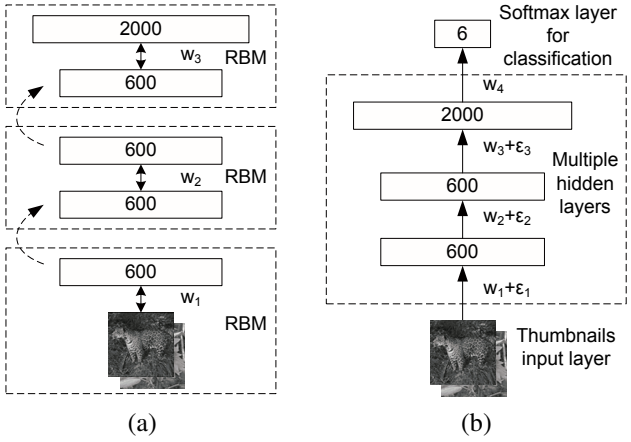

Fig. 2. (a) Stack of RBMs in Pretraining (b) Deep multi-layer networks

3.1. RBM pretraining

Pretraining learns a stack of Restricted Boltzmann Machines (RBMs) in a unsupervised way [6], as shown in Fig. 2(a). A RBM is a two-layer network where symmetric connections exist between visible units and hidden units in different layers, but there are no within-layer connections. All units are stochastic and binary. Visible units $\mathbf{v}$ correspond to observed pixels of an image, and hidden units $\mathbf{h}$ correspond to feature detectors. Let I and J be the index sets for visible and hidden units respectively, then the energy associated with a joint configuration $(\mathbf{v}, \mathbf{h})$ is given by [6]

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in I} b_i v_i - \sum_{j \in J} b_j h_j - \sum_{i,j} w_{ij} v_i h_j \quad (1)$$

where v_i and h_j are binary states of pixel i and feature j, b_i and b_j are their biases, and w_{ij} is the weight between them. The probability of the joint configuration under the model is given by

$$p(\mathbf{v}, \mathbf{h}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{Z} \quad (2)$$

where $Z = \sum_{\mathbf{u},\mathbf{g}} \exp(-E(\mathbf{u}, \mathbf{g}))$ is the partition function.

From (2) we can establish the update rules for a RBM as follows. When a pixel vector is clamped on the visible units, the hidden states are sampled from the conditional distribution $p(\mathbf{h}|\mathbf{v})$ which is factorial. Thus all hidden units can be updated in parallel, i.e., each h_j is set to 1 (h_j is on) with probability $\sigma(b_j + \sum_i v_i w_{ij})$, where $\sigma(x)$ is the logistic function $1/[1 + \exp(-x)]$. Similarly, when the binary states are chosen for the hidden units, each visible unit v_i is set to 1 with probability $\sigma(b_i + \sum_j h_j w_{ij})$.

Given a set of training images $\mathbf{v}^0$, we can first update hidden states $\mathbf{h}^0$, and then update visible states $\mathbf{v}^1$, followed by hidden states $\mathbf{h}^1$, and so on, to $\mathbf{v}^\infty$, $\mathbf{h}^\infty$, which are sampled from stationary distribution of the model. Using such alternative Gibbs sampling strategy, maximum likelihood learning can be performed. However, this process is slow and suffers from much more sampling noise. Instead, contrastive divergence learning is used[6] whereby the change in a weight only depends on the first several samplings:

$$\triangle w_{ij} = \varepsilon(< v_i^0 h_j^0 > - < v_i^1 h_j^1 >) \quad (3)$$

where ε is the learning rate, $< v_i^0 h_j^0 >$ is the fraction of times that the pixel i and feature detector j are on together when the feature detectors are being driven by data, and $< v_i^1 h_j^1 >$ is the corresponding fraction for $\mathbf{v}^1$ and $\mathbf{h}^1$. Apart from being much faster than maximum likelihood learning, it works well even though exact gradient of the log probability of the training data is not followed.

After learning a layer of feature detectors, the learned feature activations are treated as data for learning a second layer of features, i.e., the first layer of feature detectors driven by data become the visible units for learning the next RBM in the stack, as indicated by the dashed arrow below in Fig. 2(a). Such layer-wise learning can be repeated several times, and each layer of features captures strong, high-order correlations between the activities of units in the layer below. Pretraining helps generalization because it ensures that most of the information in the weights comes from modeling the thumbnails [6].

3.2. Fine-tuning and classification

After unsupervised pretraining, in order to perform object class recognition, we connect a softmax output layer to the top layer of feature extractors, as shown in Fig. 2(b). The fine-tuning stage then replaces stochastic activities of hidden layers by deterministic, real-valued probabilities and apply gradient descent to fine-tune the weights. Those weights learned by pretraining are used as initial weights in backpropagation through the whole deep neural networks. By further adjusting these weights in fine-tuning, high-order features are selected or altered for object class recognition.

Fig. 3. Examples from the image dataset of five object classes and one set of background images.

The number of softmax output units, K, is the same as the one of object classes. Each softmax unit y_k is interpreted as the probability $\hat{p}_k$ that a thumbnail falls into object class k.

$$y_k = \hat{p}_k = \frac{\exp(a_k)}{\sum_l \exp(a_l)} \quad (4)$$

where a_k is the activation for k^{th} output unit. The true class labels of training images are coded in "1-of-K" scheme, such that labels for the first class are $\mathbf{p} = (p_1, p_2, \ldots, p_K) = (1, 0, \ldots, 0)$. Then it follows naturally that the gradient descent algorithm minimizes the multi-class cross-entropy error function as given by

$$E_n = -\sum_k p_k \log \hat{p}_k \quad (5)$$

where E_n is the error for n^{th} training image.

4. EXPERIMENTS

We evaluate our approach on the image dataset previously used by Fergus et. la. for unsupervised learning[3]. It contains 5 object classes and one set of background images (Fig. 3): faces (450 images), airplanes (1074 images), cars rear (651 images), cars side (720 images), motorbikes (826 images) and background (900 images). These images have large size differences, from about 900×600 to 100×40.

First, we test our approach on 4 classes: faces, motorbikes, airplanes, and car rears, which total to more than 2800 images. 2000 images are used for training, and 800 images for testing. Three different sizes of thumbnails are considered: 30×20, 45×30, 60×40, which resulting in 600, 1350, 2400 dimensional data vectors, respectively. Raw pixel values of thumbnails are scaled to the range $[0, 1]$ for visible logistic units of the first RBM.

A network similar to Fig. 2(b) is used to train the data. There are three hidden layers and one softmx output layer.

Pretraining consists of progressively learning a stack of three RBMs. In order to speed up the pretraining stage, the data is subdivided into mini-batches, each containing 100 data vectors. Weights are updated after each mini-batch. In fine-tuning, larger mini-batches containing 1000 data vectors is used.

Apparently, the size of thumbnails is important parameter, and there is a trade-off between the degree of information preservation and computational efficiency of the learning procedure. We investigated the impact of the size on the overall performance. Different thumbnail sizes are tried, and the results are shown in Table 1. Interestingly, 60×40 achieve the same result with 45×30, although it has weights, increasing the complexity of the network. In our experiments, we use 45×30 with the corresponding network of 1350-600-600-2000-4 that achieves the best result.

size	30×20	45×30	60×40
Average accuracy	90.4	97.2	97.2

Table 1. Average accuracy on different thumbnails sizes

In order to recognize five object classes, we carry on two experiments: (E1) without background class, using a 1350-600-600-2000-5 network; (E2) with background class, using a 1350-600-600-2000-6 network. Table 4 shows the classification accuracy of our approach, compared with other state-of-the-art methods. And Table 3 shows the confusion matrix

class	(E1)	(E2)	[3]	[2]	[9]
airp	93.9	92.9	90.2	96.7	90.0
car rear	97.4	97.4	90.3	98.2	96.0
cars side	95.4	95.6	88.5	97.6	- *
faces	95.5	92.3	96.4	94.2	98.0
motorb	92.0	92.1	92.5	93.4	93.4

Table 2. Recognition results comparison in terms of accuracy. [3] learned object classes by unsupervised Scale-Invariant Learning; [2] used bag-of-words models and SVM classifiers; [9] combined kernel PCA and boosting to perform object categorization. *: Cars side class was not used in [9].

obtained with (E1). It can be seen that our approach has comparable performance. Moreover, in order to perform recognition, our approach only requires image down sampling, and passing the thumbnails through neural networks. It is a fast process, and suitable for real time implementation, since calculation in the neural networks can be carried out in parallel.

5. CONCLUSIONS

In this paper, we have presented a simple approach for object class recognition based on the intuitive idea that given only thumbnails of image, human beings are still able to recognize object classes easily. Deep neural networks are used to learn high-order correlations of thumbnails at such coarse scale. Evaluation of this approach demonstrates that it is comparable to other state-of-the-art object recognition methods. The recognition stage of our approach does not require explicit feature extraction. The proposed scheme is suitable for real time application, and easy for hardware implementation, as its workflow meets KISS principle and its associated network is structured in parallel.

True Class→	airp	cars rear	cars sides	faces	motorb
airp	199	0	0	0	6
car rear	3	222	1	2	6
cars side	0	5	103	0	1
faces	2	1	2	85	0
motorb	8	0	2	2	150

Table 3. Confusion matrix in (E1).

6. REFERENCES

[1] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[2] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan, "Categorizing nine visual classes using local appearance descriptors," in *ICPR, Workshop Learning for Adaptable Visual Systems*, 2004.

[3] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. CVPR*, 2003, vol. 2, pp. 264–271.

[4] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2003.

[5] David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2002.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," in *Proc. ICCV*, 2005, vol. 2, pp. 1792–1799.

[8] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," Tech. Rep., Dept. IRO, Universite de Montreal, 2006.

[9] S. Ali and M. Shah, "A supervised learning framework for generic object detection in images," in *Proc. ICCV*, 2005, vol. 2, pp. 1347–1354.