Learning to Recognize Hand-held Objects from Scratch

Xue Li[1,2], Shuqiang Jiang[2](✉),
Xiong Lv[2], and Chengpeng Chen[2]

[1] College of Information Science and Engineering,
Shandong University of Science and Technology, 266590 Qingdao, China;
[2] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, 100190 Beijing, China
{sqjiang}@ict.ac.cn

Abstract. Real-life environments are open-ended and dynamic: unlearned information comes over time. These changes of environments ask for the systems to have the ability of self growth. A reasonable solution is to build an intelligent human-computer interaction system to simulate the mind at birth, and then automatically teach it by human. In this work, we present a hand-held object recognition system which could incrementally enhance its recognition ability from beginning during the interaction with humans. Automatically capturing the images of hand-held objects and the voice of users, our system could refer the interacting person as a strong teacher. This allows the system to learn from scratch and to learn new concepts one after another like humans. Although our system is implemented on hand-held recognition scenario, we also implement experiments on ImageNet dataset to validate the effectiveness of our system. Experimental results illustrate its performance.

Keywords: Learn from scratch · Object recognition · SVM · Human-computer interaction

1 Introduction

Gobet *et al.* [6] propose that the configuration of smaller units of information into large coordinated units might be important in many processes of perception, learning and cognition in humans. A human baby learns everything from scratch. In addition, he can continuously learn new knowledge and build it on what he knows. This leads to a growing interest in the 'developmental' approach, which takes its inspiration from nature (especially the human infant) and attempts to build a human-computer interaction system which could develop its own knowledge and abilities through interaction with the world [7]. The idea of building an artificial baby dates back at least as far as Turing's paper on 'Computing Machinery and Intelligence' [26].

For intelligent human-computer interaction (HCI) systems, it is an important issue to learn new knowledge from scratch based on newly available data, which makes the system have the ability of self growth like a baby. Then the infant HCI system could be teached much like a human child during interaction, until it reaches an adult level. This kind of self-adjustment during interaction is a kind of on-line improvement which is timely and effectively.

In this paper, we focus on the ability of learning from scratch, by which we mean that such infant HCI systems attempt to build their own knowledge and abilities autonomously (starting with no innate knowledge), and to develop continuously to reach increasingly higher levels of knowledge. Considering an Object Recognition System (ORS), enhance its recognition ability can be regarded as a kind of self-adjustment to increase its knowledge.

Object recognition [8] is a widely studied problem in computer vision. It describes the task of finding and identifying objects in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different view points, in many different sizes and scales or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. But it is challenging for machines. Over multiple decades, many approaches have been implemented for this task. In general, an ORS has several parts. These include image sensors, image preprocessing, object detection, object segmentation, feature extraction and object classification [23].

As manipulating objects with hands is a straight way for human-machine interaction [19, 14, 15], hand-held object recognition is a special and important case in object recognition. The hand-held object can not only help the system obtain a better understanding of user's intention but also a more comprehensive perception about surrounding environment. But during interaction, the system may encounter unknown objects, which asks for the ability of self growth. Fixed models for HCI systems are unable to cope with the changes of the dynamic environments. While new concept instances will be arriving all the time, it is unrealistic and costly to retrain all previously seen images. An on-line learning HCI systems can take the advantage of incoming new concept instances to improve the existing model during interaction.

For humans to accurately understand the world around them, multi-modal integration is essential because it enhances perceptual precision and reduces ambiguity. Multi-modal sensor and feature fusion may contribute to replicate such human ability [17].

Figure 1 illustrates the procedure that our hand-held object recognition systems learn from scratch and improve its recognition ability during interaction. During interaction, our system could learn new concepts from scratch. At first, the system hasn't learn any concepts before, which means no classification model is available. It will initialize a model first, then the system can update the model with newly available objects constantly. In the process of initializing a model, our system would automatically capture the images of hand-held object as positive examples and prepare some negative examples which are not hand-held objects to train a binary classification. In later processes, the system only has to collect images of unidentified objects as positive examples. The existing model could be updated only based on these new data. RGB feature and depth feature are extracted from RGB and depth images respectively, then we fuse these two kinds of features into one.

The contributions of our recognition system can be summarized as the following:

1) Our hand-held object recognition system could learn from scratch and update the model constantly; 2) our system could automatically improve its recognition ability incrementally during interaction; 3) we validate the performance of our system on HOD and ImageNet.

The rest of this paper is organized as follows: our proposed framework is illustrated in Section 2. And the experimental results are showed in Section 3. Section 4 concludes this paper.

2 Our system

We have designed and developed a real-time self-adjustment system to learn from scratch like a human baby. Our self-adjustment framework is under the setting of SVM [27, 12]. This framework allows our system to improve the recognition ability during interaction over new obtained data without retrain all previously seen data.

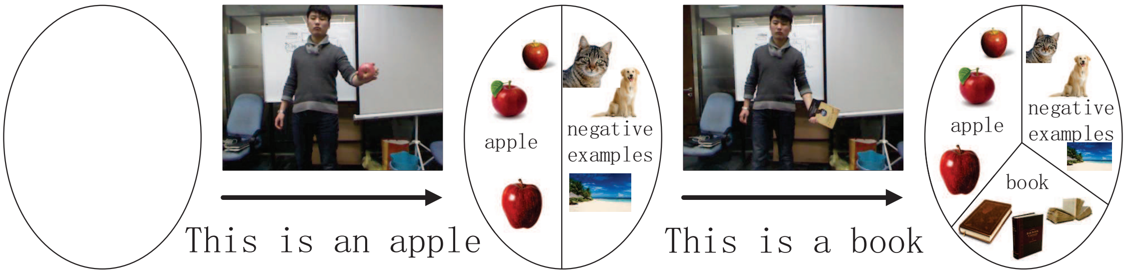


Fig. 1. The procedure that our self-adjustment hand-held object recognition system learn from scratch. At first, the system hasn't learn any concepts. It initializes a model first, and then the system updates the model with newly available objects constantly.

Object segmentation

With the surge of RGB-D devices, they provide additional depth and skeletal information, which is beneficial to eliminate background noise and make the localization and segmentation of the object easier and more precise. Our system automatically captures color and depth images using an RGB-D camera, whose API also provides additional skeletal data. Before object segmentation, the depth map is preprocessed to filter noise and recover part of the missing depth data. The system interpolates depth information in pixels where more than half of their eight neighbors have valid depth. An estimation of the location of the hand is provided in the skeletal data. We assume that hand-held objects are connected to the hand, so the hand position will typically fall in the object region.

Using the hand position as reference and initial seed, we obtain the object mask using a region-growing algorithm [15]. This algorithm examines the eight neighbors of the points in the seed set and adds them to the seed set if they are at a similar depth as the hand. The algorithm is simple yet robust. Compared with vision-based segmentation methods, the proposed method locates the target object more accurately based on the assumption.

Different from other general object or scene parsing methods, this object segmentation approach is designed specifically for the HOR problem. It utilizes the position of hands but focuses on the object held on the hand, rather than the hand itself. In HOR problem, hands remain mostly static. This approach exploits skeleton information to infer the location of the hand, and focuses on the region of hand to segment the object of interest. By exploiting prior knowledge about the HOR problem, this task-specific detection and segmentation method is more robust for HOR problem than general segmentation methods. It has three inherent advantages: background is eliminated more effectively; recognition is more reliable (as there is only one candidate); computational complexity is significantly reduced.

Feature extraction and fusion

Some hand-crafted features have been proposed to represent low-level information, like SIFT [13], spin images [10], Fast Point Features Histogram [16] and Ensembles of Shape Features (ESFs) [9]. They have been proved to be robust to transformations such as rotation and scale. But the major limitation of hand-crafted features is that they are often manually tuned for the specific conditions encountered in datasets. In addition, they can only capture a subset of the cues that are useful for recognition [28]. Except for hand-crafted features, there are also machine-learned features such as deep features. Convolutional neural networks (CNNs) [1] can learn higher order properties leading to features that can describe higher level properties of the images, which are more discriminatory [15]. Therefore, we extract features by CNNs from images. The CNN architecture that we used has eight layers: the first five layers are convolutional layers, while the sixth and the seventh layers are fully connected layers and the final layer is a softmax classifier. We use the output of the seventh layer as a feature (4096-dimensional).

There are two main types of multi-modal systems: one is to integrate signals at the feature level and the other at a semantic level [29]. For the first type, feature fusion may be beneficial to obtain a better representation of the image, and feature fusion has been used in computer vision to improve recognition accuracy. Different features often capture complementary properties of the image: RGB feature and depth feature related with color and shape respectively. Generally, feature fusion is considered more appropriate for closely coupled and synchronized modalities, such as RGB feature and depth feature [25]. There are many ways for feature fusion, we simply concatenate RGB feature and depth feature.

Image Classification

Image classification is an important and challenging task in computer vision. It includes a broad range of decision-theoretic approaches to the identification of images, such as SVM [21, 3], deep convolutional neural network [11, 22] and decision tree [5, 18]. Since image labeling is time consuming and labor intensive, image classification tasks frequently suffer from the problem of lacking sufficiency training data. So we need methods to learn incrementally from scratch.

There are some proposed approaches that could learn new information based on new available data. Ristin *et al.* [20] propose a framework named NCM Forest. They combine Nearest Class Mean classifier and Random Forest to learn new concepts. This algorithm uses hierarchical concepts to make the model can be modified locally when new data comes. Kuzborskij *et al.* [12] propose an algorithm that adds a new classification-plane to source model to learn a new concept. Michael Fink [4] proposes a one-shot learning algorithm which achieves knowledge transfer through the reuse of model parameters.

Support vector machine constructs a set of hyperplanes in feature space, which can be used for classification. The classification-planes describe the separation in feature space and support vectors describe the classification-planes. Because of these properties, it is easier to deal with the new available data for SVM.

Therefore, we extend the method of [12] to our system.

Before presenting the method, we give some preliminaries. We denote lowercase letters as vectors and capital letters as matrices. $A_{k,n}$ is corresponding to the (k, n) entry of matrix A. (x, y) represents a pair of data. x is a feature vector, and y is the corresponding label. α is denoted as Lagrange multiplier. We denote $(x_i, y_i)_{i \in I}$ as new data.

This approach is under the setting of SVM [27]. SVM solves the quadratic programming problem by minimizing an objective function, which consists of two terms: min $J_p = Risk_{espected} + Complexity$.

It utilize the multi-class LSSVM objective function [24], which turns the quadratic programming into a problem of solving linear equations. LSSVM transfers the inequality constraints to equality constraints, which greatly facilitates the solution of Lagrange multiplier method. The objective function minimizes the regularized risk by making prediction error and the complexity of this model be minimal.

$$\min_{W,b,e} J_p(W, e) = \frac{1}{2} W^T W + \frac{1}{2} C \sum\nolimits_{k=1}^{N} e_k^2 \tag{1}$$

such that

$$y_k(W^T\varphi(x_k)+b)=1-e_k,\ k=1,......,N$$

e is slack variable.

In [12], it try to find a new classification-plane w_{n+1} for unlearned concept data making N-class source model turn into a $(N+1)$-class target model. The method could achieve the following targets:

1) Finding a new group of classification-planes $W^t=[w_1,\ w_2,\cdots,\ w_n,\ w_{n+1}]$ which is close to source classification-planes making N-class source classifier transfer to a $(N+1)$-class target classifier;

2) Modifying the source classification-planes $W^{t-1}=[w_1,\ w_2,\cdots,\ w_n]$ slightly making the performance of source concepts will not decline;

3) Adjusting all source classification-planes making the prediction error be smaller.

Based on the above mentioned considerations, the objective function as following:

$$\min_{W^t,b,e} J_p(W^t,e)=\frac{1}{2}\left\|W^t-W^{t-1}\right\|^T\left\|W^t-W^{t-1}\right\|+$$

$$\frac{1}{2}\left\|w_{n+1}-W^{t-1}\beta\right\|^T\left\|w_{n+1}-W^{t-1}\beta\right\|+\frac{C}{2}\sum_{i\in I^k}(e_i)^2 \tag{2}$$

such that

$$y_i((W^t)^T\varphi(x_i)+b_i)=1-e_i,\ i\in I$$

The first term in Equation 2 controls the variety of the classification-planes of source model and this term forces the target model to keep close to the source model. The second term makes the new added classification-plane keep close to the linear combination of other existed classification-planes. The linear combination is described by a vector β. This first two terms force the target model to keep close to the source model in both situations of incremental learning, which basically guarantees the performance of other source concepts won't deteriorate. The final term is used to minimize prediction error of new data.

Finally, we can get:

$$A=A^{'}-\left[A^{''}A^{''}\beta\right]$$

$$\begin{bmatrix}A^{'}\\b^{'T}\end{bmatrix}=M\begin{bmatrix}Y\\0\end{bmatrix},\ \begin{bmatrix}A^{''}\\b^{''T}\end{bmatrix}=M\begin{bmatrix}X^TW^{'}\\0\end{bmatrix}$$

$$M=\begin{bmatrix}X^TX+\frac{1}{C} & 1\\1^T & 0\end{bmatrix}$$

The solution of this objective function is determined once we set the parameters $\beta,\ C,\ L$. The optimal β is automatically chosen by a method based on LOO error [12]. And we change C from 10^{-4} to 10^5 using 5-fold cross-validation to find the best value of C.

3 Results

In this section, we show a series of experiments to evaluate our self-adjustment system.

N	0	1	2	3	4	5	6	7	8
New Concept	-	apple	ball	book	bottle	box	calculator	can	cup
New Concept Accuracy	-	1	0.9	0.9625	0.7525	0.6775	0.7	0.74	0.78
ACC	-	1	0.95	0.9875	0.90812	0.893	0.86833	0.83143	0.84719

(a)

N	0	1	2	3	4	5	6	7	8
New Concept	-	apple	ball	book	bottle	box	calculator	can	cup
New Concept Accuracy	-	1	1	0.995	0.9575	0.8775	0.9825	0.8925	1
ACC	-	1	1	0.99833	0.97938	0.9445	0.94208	0.88857	0.9075

(b)

Table 1. N is the number of learned concepts. "New Concept" displays the new concept that the model learned in this step. "New Concept Accuracy" is the accuracy of "New Concept". And "ACC" represents the accuracy of the model. (a): incremental learning method: each column indicates a new class is introduced. (b): retraining method.

Experimental setup

We validate our system on the subset of HOD [15] and ImageNet [2]. We don't make any pre-selection or pre-processing on images in the dataset. Our datasset is splitted into two parts: train (incremental learning) and test.

Experimental results

In this section, we show the performance of our on-line learning system. Each experiment is implemented for 5 times and we show the average results.

Learn from scratch on HOD HOD consists of 16 concepts and 4 different instances for each concept. There are 200 RGB and depth image pairs for each instance. RGB feature and depth feature are extracted from RGB image and depth image respectively, and we fuse the two kinds of features into one. We randomly selected 160 images for each class considered in train datasets. And 80 images are selected for each class as a test dataset. Depth information are available in HOD, so we cascade the two kinds of features and use RBF kernel which could perform non-linear transformations of the original input features in SVM.

As Table 1(a) shows, our system learns from scratch and every step it learns 160 images of a new concept. Each column indicates a new class is introduced. Previous classes are not used for training in subsequent training steps, but they are used to evaluate the algorithm performance on previously learned classes to make sure that previously acquired knowledge was not lost. The validation on test dataset shows that our framework is able to learn the new classes, successfully. For the third columns, this step is the process of learning the first concept from scratch. In this step, our system initialize a model. And it updates this model constantly in the following steps. Because we adjust existing classification-planes while finding the new classification-plane, it can be observed that the accuracy of source concepts does not drop too much, even increases sometimes.

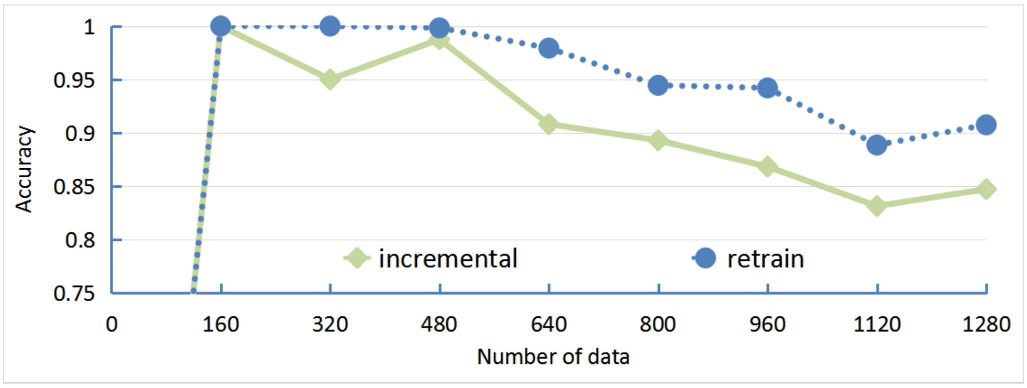
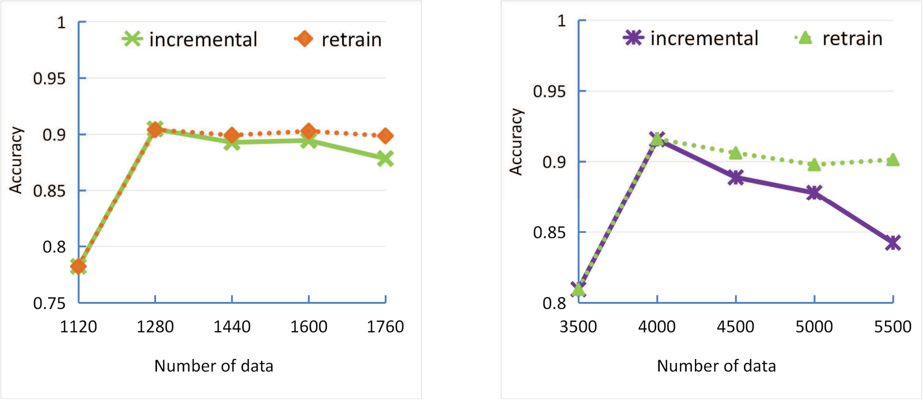


Fig. 2. Accuracies of the model: the solid line and dotted line separately represent the accuracies of incremental learning and retraining methods.

Table 1(b) shows the results of retraining. Every column means one process of retraining. Like the last column, we retrain a 8-class model. "ACC" of this column represents the accuracy this model, "New Concept Accuracy" is the accuracy of concept "cup", which is corresponding to the accuracy of new added concept in incremental learning method.

In Figure 2, the accuracy of incremental learning method is lower than retraining method. In retraining method, there are always maximum margins between concepts. But in incremental learning method, the margin between new added concept and a similar learned concept may be not maximum. There is a balance between the variety of source classification-planes and prediction error, which are corresponding to the first and last terms in Equation 2. It is the reason why the accuracy may have a slight decline after incremental learning.

Experiments in this sub section validate the ability of learning from scratch.



(a) (b)

Fig. 3. Average total accuracies: the solid line and dotted line separately represent the accuracies of incremental learning and retraining methods. (a): HOD. (b): ImageNet

N	7	8	9	10	11
New Concept	-	cup	dish	disk	glove
New Concept Accuracy	-	1	0.94	0.8425	0.905
ACC	0.77875	0.90781	0.905	0.88775	0.87295

(a)

N	7	8	9	10	11
New Concept	-	cup	dish	disk	glove
New Concept Accuracy	-	1	0.9375	0.9	0.9275
ACC	0.77875	0.90656	0.90722	0.901	0.90227

(b)

Table 2. N is the number of learned concepts. "New Concept" displays the new concept that the model learned in this step. "New Concept Accuracy" is the accuracy of "New Concept". And "ACC" represents the accuracy of the model. (a): incremental learning method: each column indicates a new class is introduced. (b): retraining method.

Statistical result on HOD In this part, we implemented our experiments on a subset of HOD which inludes eleven classes: "apple", "ball", "book", "bottle", "box", "calculator", "can", "cup", "dish", "disk" and "glove".

We split the eleven classes into two parts: one part (the former seven classes (160 imageseach class)) is used for training a source model and another part (the other four classes (160 images each class)) for incremental learning. And 80 images are selected for each class as a test dataset. We concatenate RGB feature and depth feature and use it with RBF kernel.

The source model has learned 7 source concepts ("apple", "ball", "book", "bottle", "box", "calculator", "can") and it will learn 4 new concepts ("cup", "dish", "disk", "glove") continuously in random sequences for 5 times. And each sequence we implemented for 5 times. Figure 3(a) shows the accuracies of incremental learning method (on-line self-adjustment system) and retraining method (off-line adjustment system). The accuracies of our on-line self-adjustment system are slight lower than off-line adjustment system for the reason that incremental learning method learns new information upon previously model without retraining all previously seen images. It adjusts the existing model according to the new examples instead of learning a new model. And this means that the training time of incremental learning method is much shorter than retraining method. The accuracy of the model has a slight drop during the processes of learning new concepts. This is reasonable. As the model getting complex, the accuracy decreases. Figure 3(a) shows that the accuracy in retraining method also decreases as learning new concepts continuously. For a N-class classifier, the smaller the N is, the simpler it is, and the better performance it has. And as we can see that incremental learning has smaller influence to the simpler classifier in our system. With the increasement of N, the influence of incremental learning becomes bigger and bigger to the model.

Table 2 shows an example of one random sequence of learning the 4 new concepts. Table 2(a) shows the results of incremental learning method. The model learns 160 images of new concept each step. Every adjacent two column, such as the second and third columns, they can be regarded as one process of class-incremental learning. The second column is the source model and the third is the target model. "New Concept" displays the new concept that the model learned in this process. The second column is a 7-class source model with a total accuracy of

0.7785. After learning a new concept “cup”, the source model turns into a 8-class target model with a total accuracy of 0.90781. The accuracy of new learned concept “cup” is 1.

Table 2(b) shows the results of retraining. Every column means one process of retraining. Like the third column, we retrain a 8-class model. “ACC” of this column represents the accuracy this model, “New Concept Accuracy” is the accuracy of concept “cup”, which is corresponding to the accuracy of new added concept in incremental learning method. The results show that “New Concept Accuracy” of incremental learning method are not much lower than retraining method basically.

Validations on ImageNet We implemented our experiments on a subset of ImageNet which inludes eleven classes: “orange”, “strawberry”, “coffee mug”, “pitcher”, “vase”, “plate”, “trash-can”, “envelope”, “coffeepot”, “park bench” and “lemon”.

We randomly selected 500 images each class (the former seven classes) to train a source model. And we randomly select 500 images for each unknown concepts (the last four classes) for incremental learning. And 200 images are selected for each class as a test dataset. There is no depth information available in ImageNet, so we use a single RGB feature with RBF kernel.

The source model has learned 7 source concepts (“orange”, “strawberry”, “coffee mug”, “pitcher”, “vase”, “plate”, “ashcan”) and it will learn 4 new concepts (“envelope”, “coffeepot”, “park bench”, “lemon”) continuously in random sequences. Each sequence we implemented for 5 times. Figure 3(b) shows the accuracies of incremental learning method (on-line self-adjustment system) and retraining method (off-line adjustment system). The accuracies of our on-line self-adjustment system are slight lower than off-line adjustment system.

Table 3 shows an example of one random sequence of learning the 4 new concepts. Table 3(a) shows the results of incremental learning method. The model learns 500 images of a new concept each step. Table 3(b) shows the results of retraining. Every column means one process of retraining. The results show that “New Concept Accuracy” of incremental learning method are not much lower than retraining method basically.

N	7	8	9	10	11
New Concept	-	park bench	coffee pot	lemon	envelope
New Concept Accuracy	-	0.99	0.902	0.743	0.993
ACC	0.80963	0.92025	0.88144	0.8627	0.84291

(a)

N	7	8	9	10	11
New Concept	-	park bench	coffee pot	lemon	envelope
New Concept Accuracy	-	0.99	0.902	0.873	0.98
ACC	0.80963	0.92138	0.89956	0.8861	0.89618

(b)

Table 3. N is the number of learned concepts. “New Concept” displays the new concept that the model learned in this step. “New Concept Accuracy” is the accuracy of “New Concept”. And “ACC” represents the accuracy of the model. (a): incremental learning method: each column indicates a new class is introduced. (b): retraining method.

4 Conclusion

In this work, we introduce a new on-line self-adjustment system for hand-held object recognition. The system could learn new information constantly from scratch like a human baby, which reflects the ability improve its interaction experience during interaction. It adds new classification-planes to learn unknown concepts. In the future, our on-line self-adjustment system will be extended to more recognition tasks. Besides, we will include hierarchical concepts and their relations into this framework to make human-computer interaction systems more intelligent.

Acknowledgments. This work was supported in part by the National Basic Research 973 Program of China under Grant No. 2012CB316400, the National Natural Science Foundation of China under Grant Nos. 61532018 and 61322212, the National High Technology Research and Development 863 Program of China under Grant No. 2014AA015202, Beijing Science And Technology Project under Grant No. D161100001816001. This work is also funded by Lenovo Outstanding Young Scientists Program (LOYS).

References

1. Cun, Y.L., Boser, B., Denker, J.S., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Backpropagation applied to handwritten zip code recognition. Neural Computation 1, 541–551 (1989)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
3. Doan, T.N., Poulet, F.: Large Scale Image Classification: fast Feature Extraction, Multi-CodeBook Approach and SVM Training (2014)
4. Fink, M.: Object classification from a single example utilizing class relevance metrics. In: NIPS. pp. 449–456. MIT Press (2004)
5. Friedl, M.A., Brodley, C.E.: Decision tree classification of land-cover from remotely-sensed data. Remote Sensing of Environment 61(3), 399–409 (Sep 1997)
6. Gobet, F., Lane, P., Croker, S., Cheng, P., Jones, G., Oliver, I., Pine, J.: Chunking mechanisms in human learning. Trends in Cognitive Sciences 5(6), 236–243 (2001)
7. Guerin, F.: Learning like a baby: a survey of artificial intelligence approaches. Knowledge Eng. Review 26(2), 209–236 (2011)
8. Heckerman, D.: An empirical comparison of three inference methods. CoRR (2013)
9. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: Metaxas, D.N., Quan, L., Sanfeliu, A., Gool, L.J.V. (eds.) ICCV. pp. 858–865. IEEE Computer Society (2011)
10. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell 21(5), 433–449 (1999)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) NIPS. pp. 1106–1114 (2012)
12. Kuzborskij, I., Orabona, F., Caputo, B.: From n to n+1: Multiclass transfer incremental learning. In: CVPR, 2013 IEEE Conference on. pp. 3358–3365 (June 2013)
13. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
14. Lv, X., Jiang, S., Herranz, L., Wang, S.: Hand-object sense: A hand-held object recognition system based on rgb-d information. In: Proceedings of the 23rd ACM International Conference on Multimedia. pp. 765–766. MM '15, ACM, New York, NY, USA (2015)

15. Lv, X., Jiang, S., Herranz, L., Wang, S.: Rgb-d hand-held object recognition based on heterogeneous feature fusion. Journal of Computer Science and Technology 30(2), 340–352 (2015)
16. Morisset, B., Rusu, R.B., Sundaresan, A., Hauser, K.K., Agrawal, M., Latombe, J.C., Beetz, M.: Leaving flatland: Toward real-time 3D navigation. In: ICRA. pp. 3786–3793. IEEE (2009)
17. Noda, K., Arie, H., Suga, Y., Ogata, T.: Multimodal integration learning of robot behavior using deep neural networks. Robotics and Autonomous Systems 62(6), 721–736 (2014)
18. Rajendran, P., Madheswaran, M.: Hybrid medical image classification using association rule mining with decision tree algorithm. CoRR (2010)
19. Ren, X., Gu, C.: Figure-ground segmentation improves handled object recognition in egocentric video. In: CVPR. pp. 3137–3144. IEEE Computer Society (2010)
20. Ristin, M., Guillaumin, M., Gall, J., Van Gool, L.: Incremental learning of random forests for large-scale image classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on PP(99), 1–1 (2015)
21. Sánchez, J., Perronnin, F.: High-dimensional signature compression for large-scale image classification. In: CVPR. pp. 1665–1672. IEEE Computer Society (2011)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR (2014)
23. Smith, B., Gosine, R.: Support vector machines for object recognition (2001)
24. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. 9(3), 293–300 (Jun 1999)
25. Tang, J., Jin, L., Li, Z., Gao, S.: RGB-D object recognition via incorporating latent data structure and prior knowledge. IEEE Trans. Multimedia 17(11), 1899–1908 (2015)
26. Turing, A.M.: Computing machinery and intelligence. Mind 59(236), 433–60 (Oct 1950)
27. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
28. Wang, A., Lu, J., Cai, J., Cham, T.J., Wang, G.: Large-margin multi-modal deep learning for RGB-D object recognition. IEEE Trans. Multimedia 17(11), 1887–1898 (2015)
29. Wu, L., Oviatt, S.L., Cohen, P.R.: Multimodal integration - A statistical view. IEEE Trans. Multimedia 1(4), 334–341 (1999)