Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Two-level hierarchical feature learning for image classification*

Guang-hui SONG[1,2], Xiao-gang JIN[†‡1], Gen-lang CHEN[2], Yan NIE[3]

([1]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

([2]*Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China*)

([3]*College of Science and Technology, Ningbo University, Ningbo 315100, China*)

[†]E-mail: xiaogangj@cise.zju.edu.cn

Received Oct. 20, 2015; Revision accepted Apr. 10, 2016; Crosschecked Aug. 8, 2016

Abstract: In some image classification tasks, similarities among different categories are different and the samples are usually misclassified as highly similar categories. To distinguish highly similar categories, more specific features are required so that the classifier can improve the classification performance. In this paper, we propose a novel two-level hierarchical feature learning framework based on the deep convolutional neural network (CNN), which is simple and effective. First, the deep feature extractors of different levels are trained using the transfer learning method that fine-tunes the pre-trained deep CNN model toward the new target dataset. Second, the general feature extracted from all the categories and the specific feature extracted from highly similar categories are fused into a feature vector. Then the final feature representation is fed into a linear classifier. Finally, experiments using the Caltech-256, Oxford Flower-102, and Tasmania Coral Point Count (CPC) datasets demonstrate that the expression ability of the deep features resulting from two-level hierarchical feature learning is powerful. Our proposed method effectively increases the classification accuracy in comparison with flat multiple classification methods.

Key words: Transfer learning, Feature learning, Deep convolutional neural network, Hierarchical classification, Spectral clustering

http://dx.doi.org/10.1631/FITEE.1500346 **CLC number:** TP391.4

1 Introduction

The deep convolutional neural network (CNN) has achieved impressive classification performance in the ImageNet benchmark (Krizhevsky *et al.*, 2012). Surprisingly, transfer learning methods based on the deep convolutional feature trained on a generic recognition task are also successful in various computer vision tasks, such as object classification, domain adaptation, and scene recognition. They achieve results superior to those of the previous methods (Donahue *et al.*, 2014; Zeiler and Fergus, 2014; Cai *et al.*, 2015). Therefore, the feature learning ability of deep CNN has received considerable attention. In previous studies, deep CNN models were used as feature extractors but not as classifiers, and they provided a way to obtain more specific visual features (Yosinski *et al.*, 2014).

At present, most deep CNN models serve as flat end-to-end classifiers for image recognition tasks. These deep models take the raw image as the network input, extract image features using back-propagation through layers of convolutional filters, and finally output the categorized results using a softmax output layer. However, the reality is that image datasets have a growing sample size and image category. Similarities are different among different categories, with

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61379074) and the Zhejiang Provincial Natural Science Foundation of China (Nos. LZ12F02003 and LY15F020035)

ORCID: Xiao-gang JIN, http://orcid.org/0000-0002-7787-7228

some categories being more difficult to distinguish than others. For example, let us classify three fruit varieties, i.e., apples, oranges, and bananas. The first two varieties are clearly difficult to distinguish, whereas the bananas are easy to separate from the other two. A similar situation exists in many image datasets. Fig. 1 shows that the same border colors and line types represent the highly similar categories in the Oxford Flower-102 dataset.

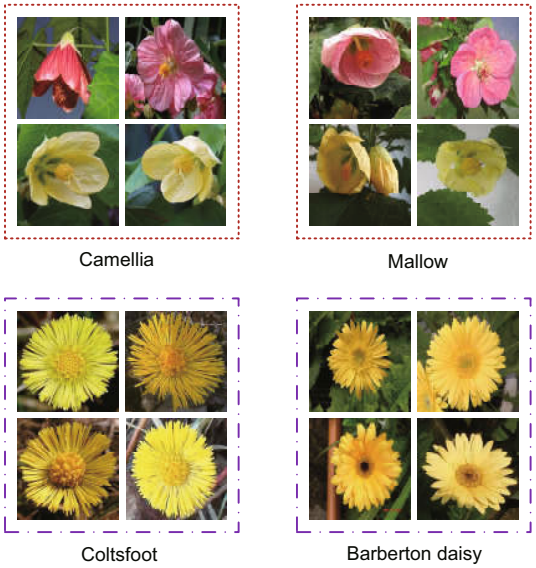

Fig. 1 The same border colors and line types representing the highly similar categories in the Oxford Flower-102 dataset. Distinguishing between camellia and mallow is difficult. The situation is similar between coltsfoot and barberton daisy. References to color refer to the online version of this figure

Inspired by the idea of transfer learning, we have come up with two questions: first, whether the general feature extractor trained on all fruit varieties can be adequate for distinguishing the categories with high similarity, and second, how to extract more specific features using the feature learning method to improve the classification performance. Adopting the hierarchical classification and feature learning methods to solve the above problem is natural. Hierarchical classification methods usually organize categories into a tree hierarchy (Deng *et al.*, 2011; Srivastava and Salakhutdinov, 2013). In this way the parent node can select different hand-designed feature descriptors according to the concrete conditions of child nodes for the classifier. In transfer learning with feature learning ability, the general features are first learned on the root node, and the specific features are further obtained from the parent nodes for the highly similar categories. Hierarchical classification methods have the advantages of high efficiency and scalability, but they can only achieve a trade-off between accuracy and efficiency (Zhao *et al.*, 2011; Liu *et al.*, 2013). In particular, too many levels produce error accumulation, which seriously affects the classification accuracy. We intend to find a way to use the hierarchical feature learning ability while preventing the accumulation of errors, leading to a decrease in accuracy. However, so far, only limited studies have focused on combining deep feature learning with hierarchical classification to improve classification accuracy (Yan *et al.*, 2015).

In this paper we propose a novel two-level hierarchical feature learning framework, which involves three steps. First, the general feature extractor is trained by fine-tuning a pre-trained deep CNN model on the target dataset. At the same time, the similarity among different categories is obtained by processing the confusion matrix. Second, the highly similar categories are divided into the same parent node, and the specific feature extractor is trained separately on each parent node. Finally, the general and specific deep features are fused into a feature vector used for the final linear classifier. The experiments using the Caltech-256, Oxford Flower-102, and Tasmania Coral Point Count (CPC) datasets demonstrate that transfer learning methods gradually enhance the expression ability of deep features. Our proposed method can effectively improve the classification accuracy based on the deep CNN models on these three datasets.

2 Related work

2.1 Deep convolutional neural network and feature learning

The deep CNN model has been widely used because of its better classification performance. It is composed of five convolutional layers (conv1–conv5), two fully connected layers (fc6 and fc7), and a softmax output layer (Krizhevsky *et al.*, 2012). This network model has strong feature learning ability and can generate feature representations by learning low-level features in the early convolutional layers and transforming them to high-level semantic features in the latter convolutional layers

(Zeiler and Fergus, 2014). Specifically, the generalization ability of the features extracted from the deep CNN model is also excellent (Donahue *et al.*, 2014). These feature representations can deal with the different tasks and datasets that have a degree of bias with respect to the ImageNet. Recently, some studies have examined transfer learning methods based on deep feature learning (Girshick *et al.*, 2014; Razavian *et al.*, 2014; Yosinski *et al.*, 2014). Transfer learning methods apply deep features to new target tasks that have too few training examples to learn full deep representations, and they gradually enhance the expression ability of deep features (Ge *et al.*, 2015). In most studies, the deep features extracted from layer fc6 or fc7 of the deep network serve as image representations (Fig. 2). In most cases, as the feature expression ability of layer fc6 is the strongest, the classification performance is the best compared to that of other layers (Donahue *et al.*, 2014).

2.2 Hierarchical classification methods

Much literature can be found on hierarchical image classification methods (Tousch *et al.*, 2012). Hierarchical image classification methods in earlier studies were designed to obtain higher classification speed at the cost of certain accuracy loss when the number of categories is very large (Deng *et al.*, 2011; Liu *et al.*, 2013). The category hierarchy is usually constructed as a tree structure, and the hierarchical tree structure may be either predefined or learned from the training data (Griffin and Perona, 2008). The method learned from the training data uses category similarity in the feature space to construct the hierarchical tree. This method includes the top-down methods (the category hierarchy is built by recursive partitioning of the set of categories) and the bottom-up methods (the category hierarchy is built by the agglomerative clustering of the categories). Some methods can also improve the classification accuracy using the known structure of the category hierarchy (Fergus *et al.*, 2010; Deng *et al.*, 2014), but better accuracy can be achieved only under some constraint conditions.

The work of Yan *et al.* (2015) is relevant to our method, but it aims to design a hierarchical end-to-end classifier based on the deep CNN to improve the classification accuracy. In Ge *et al.* (2015), a deep CNN model based on subset feature extraction was introduced. However, it is used for fine-grained category classification. Inspired by the above work, we propose a two-level hierarchical feature learning framework based on a deep CNN model. It is simple and effective, and can achieve good classification performance.

3 Approach

3.1 Hierarchical feature learning

In transfer learning methods based on deep features, a base network on a base dataset and task is trained. Then the learned deep features are repurposed to another target network to be trained on a new target dataset and task. This process will obtain better results if the features are general, i.e., suitable for both base and target tasks, instead of

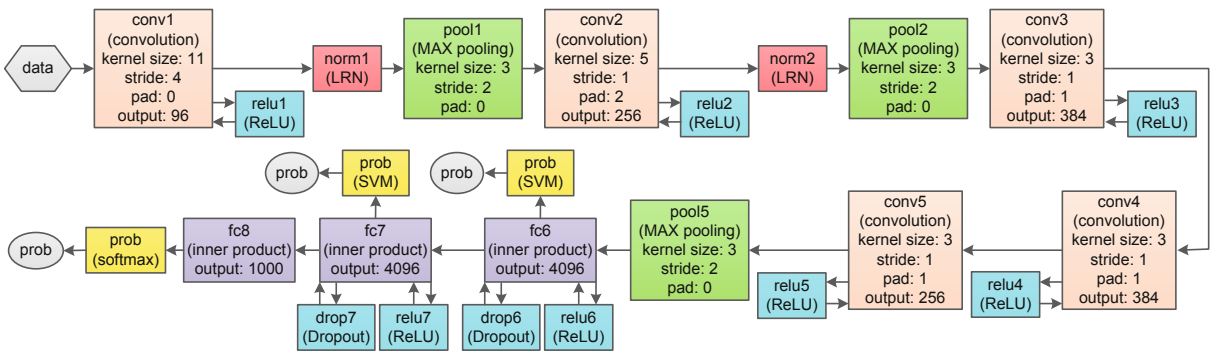

Fig. 2 Network structure of the deep CNN model. It is composed of five convolutional layers (conv1–conv5), two fully connected layers (fc6 and fc7), and a softmax output layer (Krizhevsky *et al.*, 2012). Layer fc6 or fc7 is commonly used to extract the deep features

specific to the base task. For small- and medium-scale image datasets, training a deep CNN model independently is difficult because a large number of parameters and a small number of samples will lead to an over-fitting problem. Transfer learning provides a feasible method to solve this problem. We take a pre-trained deep CNN model, modify the output category number, initialize the weights with random values on the last softmax layer, keep the remaining layers with no change, and then fine-tune the network toward the target dataset (Donahue *et al.*, 2014; Zeiler and Fergus, 2014). Better classification accuracy can be achieved if transfer learning is performed using a dataset that is the same as or related to the domains with the target dataset (Yosinski *et al.*, 2014). That is, better feature representations can be learned by progressive transfer learning, which we call hierarchical feature learning (HFL).

In our method, a target dataset is divided into two concept levels, the general dataset and the specific datasets, which are denoted as D_{g} and D_{s}, respectively. The general dataset is the whole target dataset, and the highly similar categories form the specific datasets using the clustering method. The specific dataset is a subset of the general dataset. We use a deep CNN pre-trained on the ImageNet as the base network and then fine-tune the base network to the general dataset through transfer learning. We call the fine-tuned deep CNN model the general network model, which is denoted as N_{g}. We then separately fine-tune the general network model toward the different specific datasets, which we call the specific network models, denoted as N_{s}.

The deep features of each image consist of two parts, namely general features and specific features. They are also called two-level hierarchical features. The general features are extracted from the general network model N_{g}, and the specific features are extracted from the corresponding specific network model N_{s}. The two-level hierarchical features are fused into a feature vector to enhance the expression ability of the image feature to improve classification accuracy.

3.2 General deep feature learning

We use layer fc6 of the general network model N_{g} as the general feature extractor, denoted as Φ_{g}. Similarly, layer fc6 of the specific network model N_{s} is used as the specific feature extractor, denoted as Φ_{s}. For an image $\boldsymbol{x}_i$, $\Phi_{\mathrm{g}}(\boldsymbol{x}_i)$ and $\Phi_{\mathrm{s}}(\boldsymbol{x}_i)$ represent the level-1 and level-2 deep features, respectively, extracted from the corresponding feature extractors. We can obtain the general features $\Phi_{\mathrm{g}}(\boldsymbol{x})$ of the image samples on the training set using the general network model N_{g}. The feature extracted by the deep CNN is linearly separable, and the classification accuracies of a linear and a non-linear classifier are the same basically. Since the linear support vector machine (SVM) has higher classification speed and higher classification accuracy than other classifiers, we train a linear SVM as the classifier to perform the pre-classification task. This process aims to generate a confusion matrix $\boldsymbol{M}$ for the clustering of highly similar categories.

3.3 Clustering of highly similar categories

To obtain the specific deep features, the highly similar categories should be divided into the same specific dataset. We use a top-down approach to learn the level-2 hierarchy from the validation set. Spectral clustering can be calculated so long as it is given the similarity matrix. It has many features such as better performance, low computational complexity, and better robustness. So, the spectral clustering method based on the similarity matrix $\boldsymbol{W}$ is adopted. The normalization process is conducted on the confusion matrix $\boldsymbol{M}$, which is obtained by pre-classification on all the categories. Then we transform the confusion matrix $\boldsymbol{M}$ to the similarity matrix $\boldsymbol{W}$ using the following equation:

$$\mathrm{similarity}(c_i, c_j) = \frac{1}{1 + \mathrm{distance}(c_i, c_j)}, \quad (1)$$

where c_i and c_j represent the different categories.

First, the similarity between category i and category j is measured by the distance between them. A distance matrix $\boldsymbol{L}$ is defined as $\boldsymbol{L} = \boldsymbol{1} - \boldsymbol{M}$, and its diagonal entries are set to zero. The distance matrix $\boldsymbol{L}$ is further transformed into a symmetric matrix $\boldsymbol{W}$ by the formula $\boldsymbol{W} = 0.5(\boldsymbol{L} + \boldsymbol{L}^{\mathrm{T}})$. At this point, we obtain a similarity matrix $\boldsymbol{W}$, and its entry $W_{i,j}$ is used to measure the similarity between category i and category j. Second, spectral clustering is performed on the similarity matrix $\boldsymbol{W}$ to divide the highly similar categories into k specific datasets, where k is the clustering number

to be set. According to the clustering results, we determine which categories belong to the same specific dataset D_{s_n}, $n \in \{1, 2, \ldots, k\}$. The process is described in Algorithm 1.

Algorithm 1 Two-level hierarchical feature learning training algorithm

1: Fine-tune the base network to D_g, and obtain N_g and Φ_g;
2: Train a pre-classifier, perform the pre-classification, and generate the confusion matrix $\boldsymbol{M}$;
3: Input the clustering number k, and divide the highly similar categories into the same D_s;
4: Fine-tune N_g to D_s, and obtain N_s and Φ_s;
5: Train a final classifier, and perform the prediction.

3.4 Specific deep feature learning

Transfer learning is applied by fine-tuning the general network model N_g to the k specific datasets. The specific deep feature extractors are obtained from Φ_{s_1} to Φ_{s_k}, where k is the clustering number.

To predict a test image $\boldsymbol{x}_i$, we first obtain the general deep features $\Phi_g(\boldsymbol{x}_i)$, namely the level-1 deep features. Then we conduct pre-classification using a linear SVM classifier to determine to which specific dataset D_{s_n} the image $\boldsymbol{x}_i$ belongs. The specific features $\Phi_{s_n}(\boldsymbol{x}_i)$ ($n \in \{1, 2, \ldots, k\}$), namely the level-2 deep features, are then extracted. Finally, $\Phi_g(\boldsymbol{x}_i)$ and $\Phi_{s_n}(\boldsymbol{x}_i)$ are fused into a single feature vector. We train a one-versus-rest linear SVM as the final classifier to perform the prediction task. The process is described in Algorithm 2.

Algorithm 2 Two-level hierarchical feature learning test algorithm

Input: a test image $\boldsymbol{x}_i$.
Output: the category of image $\boldsymbol{x}_i$.
1: Extract the general features $\Phi_g(\boldsymbol{x}_i)$;
2: Pre-classify image $\boldsymbol{x}_i$, and determine to which D_{s_n} image $\boldsymbol{x}_i$ belongs;
3: Extract the specific features $\Phi_{s_n}(\boldsymbol{x}_i)$;
4: Fuse $\Phi_g(\boldsymbol{x}_i)$ and $\Phi_{s_n}(\boldsymbol{x}_i)$ into a feature vector;
5: Perform the final classification, and predict the category of image $\boldsymbol{x}_i$.

4 Experiments

We evaluate our proposed method on three image datasets, namely Caltech-256 (http://www.vision.caltech.edu/Image_Datasets/Caltech256), Oxford Flower-102 (http://www.robots.ox.ac.uk/~vgg/data/flowers/102), and Tasmania CPC (http://marine.acfr.usyd.edu.au/datasets). The Caffe has been widely used as an efficient and practical deep learning framework. It is used to train deep CNN models and feature extractors in our experiments. Three deep CNN models pre-trained on ImageNet, AlexNet, CaffeNet, and VGG-16 net are used as the base models for transfer learning. The LIBLINEAR software package is used as the linear SVM classifier that adopts the one-versus-rest strategy. All the test experiments are run on a single NVIDIA Tesla K40c card. Other auxiliary algorithms, such as spectral clustering, are implemented on the Scikit-learn machine learning module based on Python.

Two evaluation measures, namely classification accuracy on all images (acc) and mean accuracy per category (mAP), are adopted in our experiments. All experiments are performed over five train/test folds.

4.1 Evaluation on Caltech-256

4.1.1 Dataset and category similarity

Caltech-256 is a general category dataset. It consists of 256 object categories and a clutter background category. It contains a total of 30 607 images for training and testing computer vision recognition and classification tasks, each category containing 80–827 images. Recently, experimental results in the literature (Donahue *et al.*, 2014; Zeiler and Fergus, 2014) have shown that transfer learning methods based on the deep CNN model improve the classification performance greatly and achieve state-of-the-art results. The mAP over five train/test folds reaches 74.2% when 60 training images per category are selected randomly. In these experiments, the deep features are extracted directly from the base deep CNN model pre-trained on the ImageNet. We call this process the CNN-base method.

By analyzing the images of the different categories, we find that a certain number of highly similar categories in visual sensation exist in the Caltech-256 dataset, such as between 71-fire-hydrant and 70-fire-extinguisher, or between 79-frisbee and 33-cd. Therefore, we evaluate our two-level HFL method following the above procedure.

4.1.2 Progressive feature learning

We randomly select 60 images per category as the training set for a total of 15 420 images. The validation set consists of 10 images per category (totally 2570 images), and the rest are used as the test set for a total of 12 617 images. The deep CNN model pre-trained on the ImageNet is fine-tuned toward the whole training set and validation set D_{g}, and N_{g} and Φ_{g} are then obtained. The initial learning rate is set to 0.0001, and it is decreased to 10% after every 15 000 iterations for a total of 40 000 iterations. The $\Phi_{\mathrm{g}}(\boldsymbol{x})$ extracted from N_{g} is used to train a linear SVM pre-classifier, and the penalty parameter C of the SVM is set to 0.03125. The pre-classification result and confusion matrix are obtained using this pre-classifier. This pre-classification process by fine-tuning the deep CNN is also expressed as the CNN-ft method.

As shown in Figs. 3a and 3b, the confusion matrix shows that the highly similar categories in visual sensation are easily misclassified. For example, six image samples of the category 89-goose are misclassified into category 60-duck, and five image samples are misclassified into category 207-swan.

Furthermore, to verify the progressive feature learning ability of transfer learning, we directly extract deep features from the base deep CNN model, denoted as the CNN-base method. Table 1 shows that the classification accuracy and mAP of the CNN-ft methods are about 0.5 percentage points higher than that of the CNN-base method in almost all of the network structure. Note that our mAP based on the base deep CNN model is lower than that of the method proposed by Zeiler and Fergus (2014) because of the differences in network structure, but it has no effect on verifying the progressive feature learning ability of transfer learning. The results show that the feature representation ability can be gradually learned using the transfer learning method.

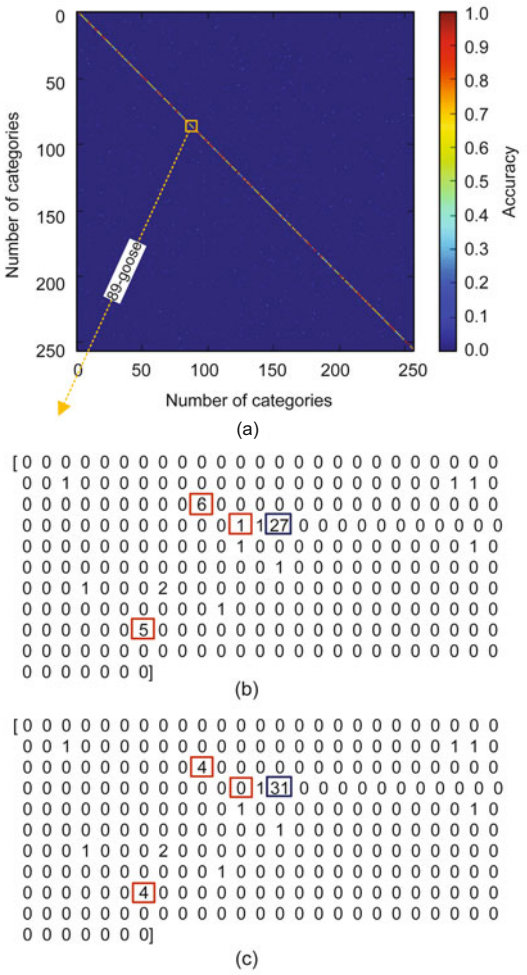

Fig. 3 The confusion matrix showing that the highly similar categories are easily misclassified: (a) the confusion matrix is generated by the result of pre-classification; (b) the classification result of category 89-goose is extracted from the confusion matrix (the number with a blue box indicates that the sample number is classified correctly and the number with a red box indicates that the sample number is misclassified); (c) the classification result of category 89-goose is extracted from the confusion matrix after using our proposed method (the sample number of misclassification among the highly similar categories is significantly reduced). References to color refer to the online version of this figure

4.1.3 Evaluation of hierarchical feature learning

We choose different clustering numbers (k) to perform spectral clustering; five k values (k = 2, 3, …, 6) are evaluated. As shown in Fig. 4, our HFL methods are effective, stable, and robust, and the classification accuracy and mAP are the best in the different network structures, as the best results increase by almost 0.6 percentage points in the VGG-16 net model when $k = 4$. At the same time, the choice of the k value is important, as it greatly affects the classification accuracy and mAP. Fig. 3c shows that the misclassification sample number of the category 89-goose is significantly reduced in the category 60-duck and 207-swan using our two-level HFL methods.

Table 1 Comparison of the performance between our two-level HFL method with optimal k value and two baseline methods on the Caltech-256 dataset

Method	acc (%)	mAP (%)
AlexNet-base (fc6)	73.17	70.87
AlexNet-ft (fc6)	73.49	71.20
AlexNet-base (fc7)	73.54	71.57
AlexNet-ft (fc7)	74.03	72.08
AlexNet-HFL (fc7, k=4)	**74.65**	**72.56**
CaffeNet-base (fc6)	73.69	71.72
CaffeNet-ft (fc6)	74.19	72.30
CaffeNet-base (fc7)	73.80	72.16
CaffeNet-ft (fc7)	74.18	72.58
CaffeNet-HFL (fc7, k=4)	**74.66**	**72.92**
VGG-16 net-base (fc6)	81.31	80.77
VGG-16 net-ft (fc6)	81.90	81.37
VGG-16 net-base (fc7)	79.41	79.69
VGG-16 net-ft (fc7)	80.70	80.61
VGG-16 net-HFL (fc6, k=4)	**82.45**	**81.90**

acc: classification accuracy on all images; mAP: mean accuracy per category. Bold numbers represent the optimal values in the corresponding test

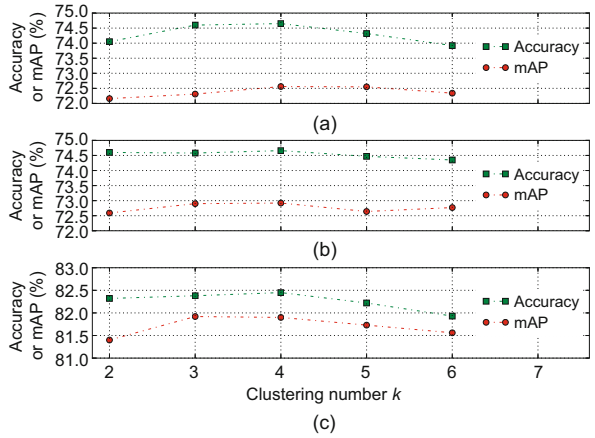

Fig. 4 Performance of our two-level HFL methods, which are stable in the different network structures on the Caltech-256 dataset: (a) AlexNet (fc7); (b) CaffeNet (fc7); (c) VGG-16 net (fc6). The choice of k greatly affects the classification accuracy and mAP

4.2 Evaluation on Oxford Flower-102

4.2.1 Dataset and baseline algorithm

Oxford Flower-102 is a fine-grained category dataset (Nilsback and Zisserman, 2008), consisting of 102 flower categories. The dataset contains a total of 8189 images, each category consisting of 40–258 images. Many methods have been tested on this dataset (Nilsback, 2009; Angelova *et al.*, 2013). In these experiments, the final performance is measured over each category, but not over all images. The measurement scale is also the mAP, which is the same as in our method for Caltech-256. The best method is the segmentation-based one, where the mAP is 80.66%, which outperforms all previous known methods (Angelova *et al.*, 2013). As shown in Fig. 1, the Oxford Flower-102 dataset contains many highly similar categories, so we evaluate our two-level HFL method based on the deep CNN on this dataset.

4.2.2 Evaluation of hierarchical feature learning

Both the training set and validation set contain 10 images per category (a total of 1020 images each). The test set consists of the remaining 6149 images. The validation set is used to fine-tune the base network, train the pre-classifier, and perform the spectral clustering. N_{g} and Φ_{g} are obtained in the same manner as for Caltech-256. The initial learning rate is set to 0.0001, and it is decreased to 10% after every 15 000 iterations for a total of 40 000 iterations. $\Phi_{\mathrm{g}}(\boldsymbol{x})$ is used to train a linear SVM pre-classifier, and the penalty parameter C is set to 0.0625. The clustering result is obtained using the confusion matrix.

As shown in Table 1, the feature extracted from layer fc6 or fc7 has mostly the same effect on the classification accuracy in these three network structures for the Caltech-256 dataset. The classification performance also becomes better with the increase of the number of network layers.

For Oxford Flower-102, the mAP obtained by using the deep CNN method outperforms that of all the previous methods (Table 2). The classification accuracy and mAP of our two-level HFL methods are the best when $k = 3$. At the same time, the feature representation ability of layer fc6 is superior to that of layer fc7. However, the better performance has not been obtained with the increase of the number of network layers. As we observe from the experimental results, each dataset has its own characteristics; the feature extracted from the right layer and a suitable network structure are both important for feature learning.

4.3 Evaluation on Tasmania CPC

4.3.1 Dataset and preprocessing

The Tasmania CPC dataset contains 1258 benthos images captured by an autonomous underwater vehicle (AUV). Each image has 50 random sample points labeled by experts. A wide range of class

labels are used that indicate biological species, abiotic elements, and types of unknown data. Fig. 5 shows an example of an image containing the experts' annotations. The precise details of the labeling methodology were described in Barrett *et al.* (2011). The different-sized patches are evaluated to achieve better classification performance. A total of 62 900 data points from 1258 images are included in the dataset. However, the distribution of sample instances is highly unbalanced. For example, the 'soft' category has 28 569 instances, whereas the 'hard' category has only 144 instances. In this case, the data reconstruction approach based on over-sampling and under-sampling is introduced. Specifically, for over-sampling, we obtain different-sized patches in a certain size range to compensate for the shortage of the samples. Finally, we extract eight categories and 21 092 sample instances to construct the target dataset D_g. The sample number of each category is relatively balanced in the general dataset. We randomly extract two thirds of the sample images per category as the training set and validation set, and the remaining is used as the test set.

Table 2 Comparison of the performance between our method with the optimum k value and two baseline methods on the Oxford Flower-102 dataset

Method	acc (%)	mAP (%)
AlexNet-base (fc6)	79.62	81.55
AlexNet-ft (fc6)	80.47	82.58
AlexNet-base (fc7)	76.51	78.70
AlexNet-ft (fc7)	78.59	80.74
AlexNet-HFL (fc6, k=3)	**81.04**	**83.16**
CaffeNet-base (fc6)	79.24	81.54
CaffeNet-ft (fc6)	80.42	82.68
CaffeNet-base (fc7)	76.48	79.01
CaffeNet-ft (fc7)	79.33	81.46
CaffeNet-HFL (fc6, k=3)	**81.38**	**83.35**
VGG-16 net-base (fc6)	78.66	80.58
VGG-16 net-ft (fc6)	79.82	81.24
VGG-16 net-base (fc7)	74.56	76.64
VGG-16 net-ft (fc7)	77.27	78.36
VGG-16 net-HFL (fc6, k=3)	**80.21**	**81.86**

acc: classification accuracy on all images; mAP: mean accuracy per category. Bold numbers represent the optimal values in the corresponding test

4.3.2 Patch sizes and deep features

The experts determine the category labels for each random sample point. Therefore, our goal is to classify local image patches centered around the sample points. The deep features of the images in the experiments are extracted from the local image patches. Square patches of different sizes are used. A range of scales are evaluated from 60×60 pixels to 160×160 pixels.

For the Tasmania CPC dataset, we choose only the CaffeNet model to perform the evaluation. We fine-tune this model using the transfer learning method on the general dataset D_g, and obtain N_g and Φ_g. Each size of the patches corresponds to a general feature extractor. The initial learning rate of the network is set to 0.0001, and it is decreased to 10% after every 12 000 iterations for a total of 30 000 iterations. The general features $\Phi_g(\boldsymbol{x})$ extracted from N_g are used as inputs of the linear SVM classifier for training a pre-classifier. The output

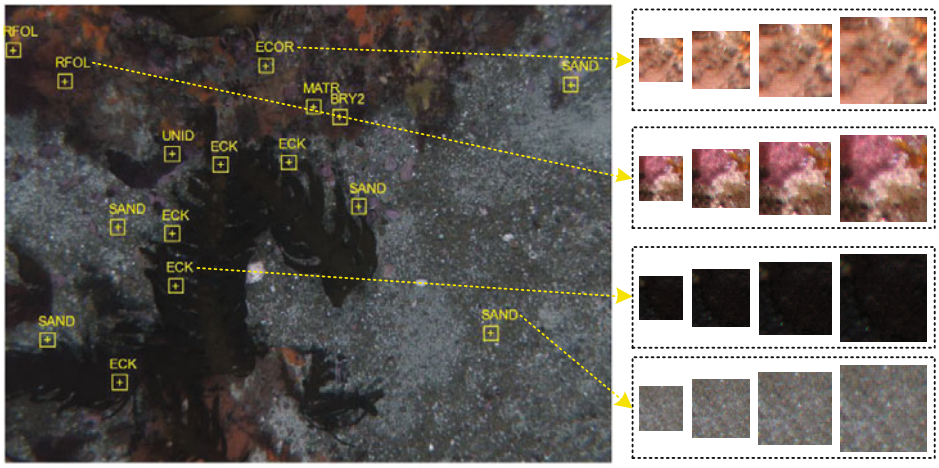

Fig. 5 An example of an autonomous underwater vehicle (AUV) image containing the experts' annotations. The different-sized patches are evaluated to achieve better classification performance from 60×60 to 160×160 pixel patches

results of the pre-classifier are used to evaluate the classification performance of different-sized patches. The penalty parameter C of the SVM classifier is set to 0.0625. The pre-classification process is called the CaffeNet-ft method for comparison with other classification methods.

As shown in Table 3, the classification performance of the CaffeNet-ft method is the best. The mAP reaches 92.11% when the patch size is 120×120 pixels. In the following experiments, we use the experimental results on the 120×120 pixel patches as the benchmark for comparison with other methods.

Table 3 Comparison of classification performance based on deep features and conventional hand-designed features in various sizes patches

Method	Size (pixels)	mAP (%)
PCA	63×63	80.13
GLCM	95×95	74.82
LBP	31×31	71.35
CaffeNet-ft (fc6)	60×60	89.90
	80×80	90.59
	100×100	91.24
	120×120	**92.11**
	140×140	91.51
	160×160	91.55

The mAP averaging over five data splits is used as the measurement scale. PCA: principal component analysis; GLCM: gray-level co-occurrence matrix; LBP: local binary pattern. mAP: mean accuracy per category. The bold number represents the optimal value

4.3.3 Comparison of different feature extraction methods

In this subsection, we compare the effectiveness of deep convolutional features and conventional hand-designed features. Bewley *et al.* (2012; 2015) have analyzed the performance of principal component analysis (PCA), local binary pattern (LBP) texture descriptors, and gray-level co-occurrence matrix (GLCM) on the Tasmania CPC dataset. We extract the hand-designed features from local image patches around the ground truth label using the above methods. We also use a linear SVM classifier to evaluate the classification effect. The best performances of the three methods in the appropriate patch size are selected to compare with those of the deep feature methods.

As shown in Table 3, the worst performance based on the deep feature learning method is better than that based on the hand-designed feature descriptors. This shows that deep features and transfer learning methods are effective on the Tasmania CPC dataset.

4.3.4 Performance of hierarchical feature learning

We choose the 120×120 pixel patches to perform the two-level HFL experiments. To build the category hierarchy, we first perform a pre-classification on the validation set and calculate the similarity matrix $\boldsymbol{W}$ of all the categories. The clustering number k is set to 2, 3, and 4. We perform spectral clustering to obtain highly similar categories to form the k specific datasets. Then we train the specific feature extractor Φ_{s_n} $(n \in \{1,\ 2,\ \ldots,\ k\})$ using the transfer learning method. Finally, we fuse $\Phi_{\mathrm{g}}(\boldsymbol{x}_i)$ and $\Phi_{\mathrm{s}_n}(\boldsymbol{x}_i)$ into an 8912-dimensional feature vector and use it as an image feature representation for training the final classifier. We compare the classification performance of our proposed method with that of the CaffeNet-ft method on the testing set.

As shown in Table 4, the mAP is up to 93.76% using the feature of layer fc6 when $k = 2$. However, if the k is not appropriately selected, the classification performance will be compromised, e.g., $k = 4$. The experiments demonstrate that the feature learning method fine-tuned on the deep CNN model can be applied to the Tasmania CPC dataset.

Table 4 Performance of our proposed method with different k values (patch size: 120×120 pixels)

Method	mAP (%)
CaffeNet-ft	92.11
CaffeNet-HFL (fc6, k=2)	**93.76**
CaffeNet-HFL (fc6, k=3)	92.52
CaffeNet-HFL (fc6, k=4)	92.02

mAP: mean accuracy per category. The bold number represents the optimal value

5 Conclusions

In this paper, we proposed a two-level HFL framework based on transfer learning to solve the misclassification problem in highly similar categories. The specific deep features are gradually learned using the two-level HFL method to improve the classification performance. The experimental results indicate that our method is better than the baseline methods on the Caltech-256, Oxford Flower-102, and Tasmania CPC datasets,

as it has higher classification accuracy and mAP. Moreover, the choice of the clustering number k has a great impact on the classification performance in the proposed method. When k is not appropriate, the classification accuracy is even lower than that of the baseline method. In the future, we will conduct the experiments with a large-scale dataset to verify the adaptability of the proposed method. We will also explore a more suitable clustering approach for highly similar categories to further improve the classification accuracy.

Acknowledgements

The authors would like to acknowledge the Australian National Research Program (NERP) Marine Biodiversity Hub for the taxonomical labeling and the Australian Center for Field Robotics for gathering the image data.

References

Angelova, A., Zhu, S., Lin, Y., 2013. Image segmentation for large-scale subcategory flower recognition. Proc. IEEE Workshop on Applications of Computer Vision, p.39-45. http://dx.doi.org/10.1109/WACV.2013.6474997

Barrett, N., Meyer, L., Hill, N., *et al.*, 2011. Methods for the Processing and Scoring of AUV Digital Imagery from South Eastern Tasmania. Technical Report. University of Tasmania, Hobart.

Bewley, M.S., Douillard, B., Nourani-Vatani, N., *et al.*, 2012. Automated species detection: an experimental approach to kelp detection from sea-floor AUV images. Proc. Australasian Conf. on Robotics and Automation, p.1-10.

Bewley, M.S., Nourani-Vatani, N., Rao, D., *et al.*, 2015. Hierarchical classification in AUV imagery. Proc. 9th Int. Conf. on Field and Service Robotics, p.3-16. http://dx.doi.org/10.1007/978-3-319-07488-7_1

Cai, Y., Yang, M.L., Li, J., 2015. Multiclass classification based on a deep convolutional network for head pose estimation. *Front. Inform. Technol. Electron. Eng.*, **16**(11):930-939. http://dx.doi.org/10.1631/FITEE.1500125

Deng, J., Satheesh, S., Berg, A.C., *et al.*, 2011. Fast and balanced: efficient label tree learning for large scale object recognition. Advances in Neural Information Processing Systems 24, p.567-575.

Deng, J., Ding, N., Jia, Y.Q., *et al.*, 2014. Large-scale object classification using label relation graphs. Proc. 13th European Conf. on Computer Vision, p.48-64. http://dx.doi.org/10.1007/978-3-319-10590-1_4

Donahue, J., Jia, Y., Vinyals, O., *et al.*, 2014. DeCAF: a deep convolutional activation feature for generic visual recognition. Proc. 31st Int. Conf. on Machine Learning, p.1-9.

Fergus, R., Bernal, H., Weiss, Y., *et al.*, 2010. Semantic label sharing for learning with many categories. Proc. 11th European Conf. on Computer Vision, p.762-775. http://dx.doi.org/10.1007/978-3-642-15549-9_55

Ge, Z.Y., McCool, C., Sanderson, C., *et al.*, 2015. Subset feature learning for fine-grained category classification. Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, p.46-52. http://dx.doi.org/10.1109/CVPRW.2015.7301271

Girshick, R., Donahue, J., Darrell, T., *et al.*, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.580-587. http://dx.doi.org/10.1109/cvpr.2014.81

Griffin, G., Perona, P., 2008. Learning and using taxonomies for fast visual categorization. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.1-8. http://dx.doi.org/10.1109/cvpr.2008.4587410

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25, p.1097-1105.

Liu, B.Y., Sadeghi, F., Tappen, M., *et al.*, 2013. Probabilistic label trees for efficient large scale image classification. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.843-850. http://dx.doi.org/10.1109/CVPR.2013.114

Nilsback, M.E., 2009. An Automatic Visual Flora—Segmentation and Classification of Flower Images. PhD Thesis, University of Oxford, UK.

Nilsback, M.E., Zisserman, A., 2008. Automated flower classification over a large number of classes. Proc. 6th Indian Conf. on Computer Vision, p.722-729. http://dx.doi.org/10.1109/ICVGIP.2008.47

Razavian, A.S., Azizpour, H., Sullivan, J., *et al.*, 2014. CNN features off-the-shelf: an astounding baseline for recognition. Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, p.512-519. http://dx.doi.org/10.1109/CVPRW.2014.131

Srivastava, N., Salakhutdinov, R.R., 2013. Discriminative transfer learning with tree-based priors. Advances in Neural Information Processing Systems 26, p.2094-2102.

Tousch, A.M., Herbin, S., Audibert, J.Y., 2012. Semantic hierarchies for image annotation: a survey. *Patt. Recogn.*, **45**(1):333-345. http://dx.doi.org/10.1016/j.patcog.2011.05.017

Yan, Z.C., Zhang, H., Piramuthu, R., *et al.*, 2015. HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. Proc. Int. Conf. on Computer Vision, p.2740-2748.

Yosinski, J., Clune, J., Bengio, Y., *et al.*, 2014. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems 27, p.3320-3328.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. Proc. 13th European Conf. on Computer Vision, p.818-833. http://dx.doi.org/10.1007/978-3-319-10590-1_53

Zhao, B., Li, F., Xing, E.P., 2011. Large-scale category structure aware image categorization. Advances in Neural Information Processing Systems 24, p.1251-1259.