Improving Image Classification Performance with Automatically Hierarchical Label Clustering

Zhiqiang Chen[1,2], Changde Du[1,2], Lijie Huang[1], Dan Li[1,2], Huiguang He[1,2,3*]
[1]Research Center for Brain-inspired Intelligence and National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, China
{chenzhiqiang2014, duchangde2016, lijie.huang, lidan2017, huiguang.he}@ia.ac.cn
*Corresponding author

Abstract—**Image classification is a common and foundational problem in computer vision. In traditional image classification, a category is assigned with single label, which is difficult for networks to learn better features. On the contrary, hierarchical labels can depict the structure of categories better, which helps network to learn more hierarchical features and improve the classification performance. Though many datasets contain images with multi-labels, the labels in these datasets usually lack of hierarchy. To overcome this problem, we propose a new method to improve image classification performance with Automatically Hierarchical Label Clustering (AHLC). Firstly, AHLC calculates the similarity between each pair of original categories by how easily they are misclassified with a pre-trained classifier. Secondly, AHLC obtains hierarchical labels by merging similar categories using hierarchical clustering. Finally, AHLC trains a new classifier with hierarchial labels to improve the original classification performance. We evaluate our method on MNIST and CIFAR-100 datasets and the results demonstrate the superiority of our method. The main contribution of this work is that we can simply improve an existing classification network by AHLC without extra information or heavy architecture redesign.**

I. Introduction

In the last decade, the capabilities of object classification have been dramatically improved due to advances in deep learning and convolutional nerual networks (CNN) [1]. In the early stage of CNN development, LeNet [2] only had 3 convolution layers and 2 fully connected layers, and showed powerful performance in handwriting digits recognization. To obtain larger classification capabilities, on the one hand, deeper architecture such as VGG [3], GoogleNet [4], ResNet [5] have been developed, each of which made great breakthrough. On the other hand, some effective functional layers have been exploited such as ReLU [6], dropout [7], batch normalization [8] and became common operations.

Many researchers concentrated on classifier design, and usually neglected the way we label images, which is a brief description of image. In the early studies of classification, one image had exactly one label. When the scale of dataset grows, unique label is difficult to depict the differences between categories. For example, the widely used ImageNet dataset with 1000 labels, which has brought great breakthrough to the development of computer vision, can't depict images well because of being too similar of some labels. In ImageNet dateset, Maltese dog and Pekinese are two categories, which are two kinds of dog, and they are very similar. While school bus is also a category and it's much different from dog. So an unique label can't depict the differences of categories well. To characterize the differences, hierarchical labels are more natural and reasonable, which are consistent with human cognition. Though there are many studies on multi-label learning, its labels usually lack of hierarchy and need to be labeled manually [9–11]. In practice, it's difficult to label massive images with hierarchical labels because different people has different criteria. Zhu et al. [12] tried to merge the similar labels, and it provided a new idea to generate new labels without heavily manually labeling. However, Zhu et al. [12] did't give attention to hierarchical labels, which couldn't benefit from hierarchical features learned from hierarchical labels. In traditional machine learning, hierarchical label learning has been explored by some classical methods such as decision tree [13, 14]. However, in the field of deep learning, there are few studies focusing on hierarchical label learning to boost the classification performance.

In this paper, we propose a method to improves the performance of classification with automatically hierarchical label clustering (AHLC). AHLC consists of three stages: (1) Calculate the similarity between each pair of original categories by how easily they are misclassified by a pre-trained classifier; (2) Obtain hierarchical labels by merging similar categories using hierarchical clustering; (3) Train a new classification network with hierarchical labels obtained by stage 2. The proposed method AHLC requires no extra labeling information or heavy architecture redesign of network. It can be easily applied to existing excellent classification networks.

II. Related Works

LeNet-5 [2], which starts the study of CNN, showed an impressive performance compared with traditional machine learning methods in handwriting digits classification. To improve the performance of CNN, AlexNet [15] trained a deep CNN, which used activation function ReLU instead of traditional sigmoid to speed up convergence. To investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting, VGG [3] conducted

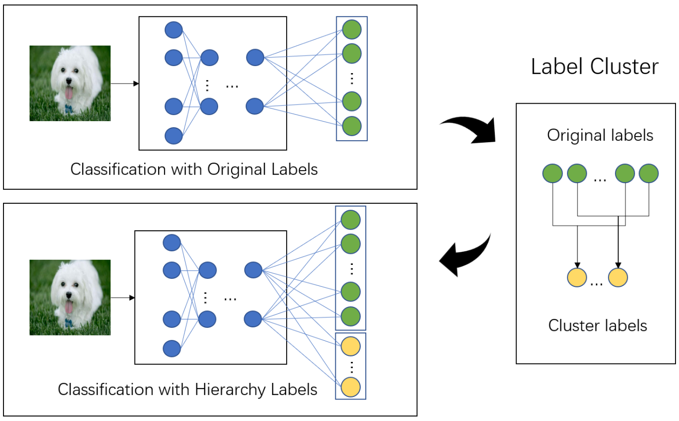
Classification with Original Labels
Label Cluster
Original labels
Cluster labels
Classification with Hierarchy Labels

Fig. 1. Illustration of our proposed AHLC. There are three stages in AHLC: 1) A normal classification network is trained with original labels to calculate similarity between labels; 2) Obtain hierarchical labels by clustering with the similarity calculated in stage 1; 3) Train new classification network with hierarchical labels.

a thorough evaluation of networks of increasing depth, and it showed that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16−19 layers. In GoogleNet [4], inception module was proposed to combine the features of different receptive field filters and it achieved the state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge (ILSVR) 2014. With residual block, He et al. [5] came up with ResNet with an extreme depth of 152 and won 1st place on almost all major tasks in ILSVRC 2015 and Microsoft Common Objects in Context 2015 competitions.

Zhang et al. [9] also noticed that single label was difficult to describe an image, as real-world objects might be complicated and have multiple semantic meanings simultaneously. Multi-label learning [9–11] labels an image by multi labels. In [16, 17], they introduced a joint patch and multi-label learning framework that models the structured joint dependence behind features. Labeling an image with multi-label manually consumes heavy workload, Zhu et al. [12] merged similar labels to generate new labels. Labels in Multi-label learning always lacked of hierarchy, while in traditional machine learning, [13, 14] can naturally obtain a hierarchical architecture and perform classification.

III. Proposed Method

In this section, we present the proposed method AHLC to improve the performance of classification with Automatically Hierarchical Label Clustering. As shown in Fig. 1, AHLC contains three stages: Firstly, train a classification network with the original labels and calculate the similarity between different labels; Secondly, cluster the original labels into hierarchical labels by merging similar labels via the similarity; Finally, train a new classification network with hierarchical labels to gain a further improvement of performance for original classification task.

A. Calculate similarity between labels

Ideally, we can label categories with hierarchical labels manually, which is time consuming and require expert knowledge. What is worse, most of the public datasets don't have hierarchical labels, so it's meaningful to cluster hierarchical labels automatically. To cluster the original labels into hierarchical labels automatically, we need to get the similarity between different labels. In an indirect way, we can evaluate it according to how easily two labels are misclassified. So in this stage, we train a classification network with original labels.

For image set $\mathbf{X} = \{x_1, x_2, ..., x_m\}$ and label set $\mathbf{L} = \{l_1, l_2, ..., l_n\}$, m and n are the number of images in image set, labels in label set respectively. A discriminative network f is trained to classify the original categories. For any $x \in \mathbf{X}$, the prediction $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ can be obtained by

$$\boldsymbol{y} = f(x), y_i = f_i(x), i \in \{1, 2, ..., n\}, \tag{1}$$

where y_i is the prediction of label l_i.

After the classifier f is trained, we use the prediction of f to statistic the probabilities of each pair labels misclassified one for another.

Specifically, for any $x_p \in \mathbf{X}$ corresponded with label $Y = l_i \in \mathbf{L}$, predict label by networks $P = l_j \in \mathbf{L}$ is converted to a probability distribution format by a softmax function as

$$\Pr(P = l_j | Y = l_i, X = x_p) = \frac{e^{f_j(x_p)}}{\sum_{k=1}^{n} e^{f_k(x_p)}}. \tag{2}$$

The probability of $\Pr(P = l_j | Y = l_i)$ indicates that predicting label is l_j in the condition that the real label is l_i, conveniently marked as $\Pr(l_j|l_i)$. We can obtain $\Pr(l_j|l_i)$ by

$$\begin{aligned}
&\Pr(l_j|l_i)\\
=&\Pr(P = l_j | Y = l_i)\\
=&\frac{\Pr(P = l_j, Y = l_i)}{\Pr(Y = l_i)}\\
=&\frac{\sum_{p=1}^{m} \Pr(P = l_j, Y = l_i, X = x_p)}{\Pr(Y = l_i)}\\
=&\frac{\sum_{p=1}^{m} \Pr(P = l_j | Y = l_i, X = x_p) * \Pr(Y = l_i, X = x_p)}{\Pr(Y = l_i)}.
\end{aligned} \tag{3}$$

So we can naturally define the similarity between label l_i and l_j by simply summing $\Pr(l_j|l_i)$ and $\Pr(l_i|l_j)$. Category can't be merged with itself, so we define the similarity between a category with itself as zero. So the similarity S is defined as

$$S(l_i, l_j) = \begin{cases} \Pr(l_j|l_i) + \Pr(l_i|l_j), & \text{for } \ l_i \neq l_j \\ 0, & \text{for } \ l_i = l_j. \end{cases} \tag{4}$$

Labels would be merged if we applied cluster algorithm, so we need define the similarity between original label and merged label. An original label is similar with a merged label if it's similar with either label in merged label, so we take the maximum similarity of the two labels. Specifically, suppose

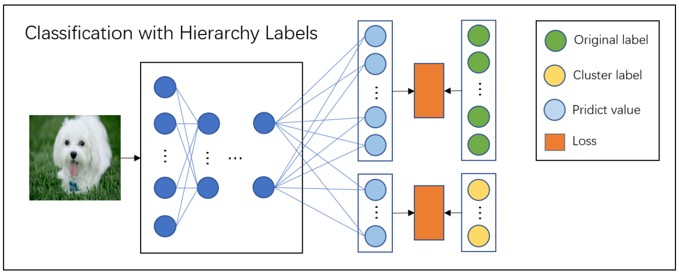

Fig. 2. Architecture of classification with hierarchical labels.

we merge label l_i and l_j as $M(l_i, l_j)$, the similarity between l_k and the merged label $M(l_i, l_j)$ is defined as

$$S(l_k, M(l_i, l_j)) = \max(S(l_k, l_i), S(l_k, l_j)). \tag{5}$$

B. Label Cluster

In this stage, AHLC aims to obtain hierarchical labels by the similarity calculated by stage 1. Via the similarity measure S, we apply hierarchical cluster method [18] which is one of the most widely used cluster method, to cluster labels to a certain number.

Algorithm 1 shows details how we get hierarchical labels. For original labels with n_1 labels and cluster labels with expected number n_2, do $n_1 - n_2$ times merge operations, which merges the most similar two labels in each time. Then we can obtain n_2 cluster labels.

Algorithm 1 Hierarchical Cluster

```
Inputs:
    L: original labels
    $S$: the similarity between any two labels in L
    $n_1$: the number of original labels
    $n_2$: the number of cluster labels
Outputs:
    $\mathbf{L}^{'}$: cluster labels
Ensure:
    $\mathbf{L}^{'} \leftarrow \mathbf{L}$
    for $t = 1$ to $n_1 - n_2$ do
        find $l_i$ and $l_j$ in $\mathbf{L}^{'}$ with maximum similarity $S(l_i, l_j)$, according to Eq. (4) and Eq. (5)
        merge $l_i$ and $l_j$ and update $\mathbf{L}^{'}$
    end for
    return $\mathbf{L}'$
```

C. Classification with Hierarchical Labels

As we obtain hierarchical labels which consist of original labels and cluster labels, we add a new branch for cluster labels at the prediction layer base on baseline network. It's a general method to improve classification performance without adding extra information and heavy network structure redesign. Fig. 2 illustrates the architecture of AHLC, which increases an extra branch with cluster labels based on original classification networks.

Specifically, original label set has n_1 labels and cluster label set has n_2 labels. For an instance image x with original label $\boldsymbol{l}_o$ and cluster label $\boldsymbol{l}_c$ whose length are n_1 and n_2 respectively, which are both one-hot. For predict values of original labels $\boldsymbol{y}_o = (y_{o1}, y_{o2}, ..., y_{on_1})$ and predict values of cluster labels $\boldsymbol{y}_c = (y_{c1}, y_{c2}, ..., y_{cn_2})$, we apply a softmax function for them and get $\boldsymbol{y}_o^{'}$ and $\boldsymbol{y}_c^{'}$:

$$y_{oi}^{'} = \frac{e^{y_{oi}}}{\sum_{j=1}^{n_1} e^{y_{oj}}}, y_{ci}^{'} = \frac{e^{y_{ci}}}{\sum_{j=1}^{n_2} e^{y_{cj}}}. \tag{6}$$

To get losses of the AHLC, a cross entropy loss H is used:

$$\mathrm{H}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{i=1}^{n} y_i * log(x_i), \tag{7}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are predict values and one-hot labels respectively and n is the length of $\boldsymbol{x}$ and $\boldsymbol{y}$.

To improve the performance of classification by multi-task, we add losses of original labels and cluster labels in total as L. To better take advantages of promotion of multi-task and reduce the influence of original task from newly adding task, a weight decay λ_c is applied for cluster labels task, which is expected less than 1 because the main task is to improve the performance of original task. Under l_2 regularization, whole loss L is

$$L = \mathrm{H}(\boldsymbol{y}_o^{'}, \boldsymbol{l}_o) + \lambda_c * \mathrm{H}(\boldsymbol{y}_c^{'}, \boldsymbol{l}_c) + \lambda_w * \|W\|_2^2, \tag{8}$$

where W are weights in total networks, and λ_w is coefficient of l_2 regularization.

IV. Experiments and Results

This section focuses on the evaluation of the proposed method by measuring the improvements brought by classification with hierarchical labels compared with original labels. To quantify the effect of the proposed method, we perform an ablation study on MNIST and CIFAR100, which are both excellent datasets widely used by researchers of computer vision and artificial intelligence. In our experiments, all training strategies and parameters of baseline networks and proposed AHLC are the same, and the structures are also the same expect an additional predict branch for cluster labels at the end of AHLC. To compare the performance of baseline networks and proposed AHLC, the accuracy of original labels instead of hierarchical labels are calculated.

A. MNIST

The MNIST database[1] of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples, where the digits have been size-normalized and centered in a fixed-size image. Training is performed on 28×28 images that have been shifted by up to 2 pixels in each direction with

[1] http://yann.lecun.com/exdb/mnist/

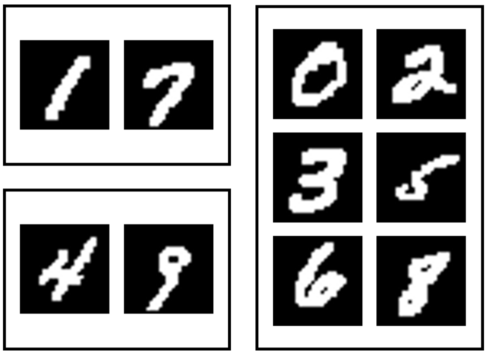

Fig. 3. Cluster result of MNIST. The original 10 labels is clustered into 3 layers. The first cluster label contains the original labels 1, 7, the second contains 4, 9 and the third contains 0, 2, 3, 5, 6, 8. The images are randomly chosen in each original labels.

TABLE I
AHLC CLASSIFICATION TEST ACCURACY ON MNIST DATASET. TRAINING IS PERFORMED ON 28 × 28 IMAGES THAT HAVE BEEN SHIFTED BY UP TO 2 PIXELS IN EACH DIRECTION WITH ZERO PADDING. NO OTHER AUGMENTATION/DEFORMATION IS USED. WE TRAIN A BASELINE NET WITH ORIGINAL LABELS TO COMPARE WITH PROPOSED AHLC. OUR BASELINE NET IS A STANDARD CNN WITH THREE CONVOLUTIONAL LAYERS OF 256, 256, 128 CHANNELS [19]. EACH HAS 5 × 5 KERNELS AND STRIDE OF 1. THE LAST CONVOLUTIONAL LAYERS IS FOLLOWED BY TWO FULLY CONNECTED LAYERS OF SIZE 328, 192. THE LAST FULLY CONNECTED LAYER IS CONNECTED WITH DROPOUT TO A 10 CLASS SOFTMAX LAYER WITH CROSS ENTROPY LOSS.

method	BaseLine	AHLC-e	**AHLC(proposed)**
error(%)	0.39	0.40	**0.36**

zero padding. No other augmentation/deformation is used. We train a baseline net with original labels to compare with proposed AHLC. Our baseline net is a standard CNN with three convolutional layers of 256, 256, 128 channels [19]. Each has 5×5 kernels and stride of 1. The last convolutional layers is followed by two fully connected layers of size 328, 192. The last fully connected layer is connected with dropout to a 10 class softmax layer with cross entropy loss.

We set the cluster number as 3. Then the cluster labels is shown in Fig. 3. The original labels 1 and 7 are merged into same cluster. Similarly, 4 and 9 are merged and others are merged.

With the hierarchical labels, we add a new branch for cluster labels and train AHLC by loss L defined as Eq. 8. And AHLC sets λ_c as 0.5 to make the original task dominant one. AHLC-e sets λ_c as 1 to treat tasks of original labels and cluster labels equally. We adopt weight decay 0.0002 of l2 regularization for both baseline network and AHLC.

Table I illustrates accuracy on test set and Fig. 4 illustrates classification accuracy of original task vs. training batches. AHLC has a lowest test error and 7.7% lower than baseline network. AHLC-e has a slightly higher test error than baseline network.

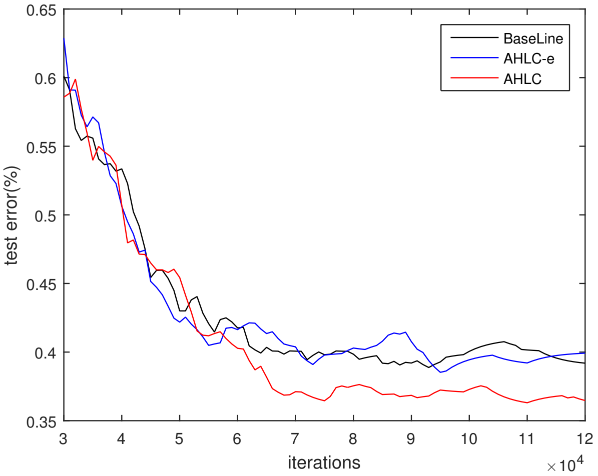

Fig. 4. AHLC classification test accuracy on MNIST vs. training batches under a moving average smooth of 0.2.

B. CIFAR-100

CIFAR dataset[2] is an established computer vision dataset used for object recognition, representing a great starting point towards future applications. CIFAR contains CIFAR-10 and CIFAR-100 and we choose CIFAR-100 as our experimental dataset for its rich labels. The CIFAR-100 dataset consists of 60000 32x32 colour images in 100 classes, with 600 images per class, 500 for training images and 100 for test images. Besides the 100 fine labels, CIFAR-100 also has 20 coarse labels, which divides the 100 fine labels into 20 labels and each contains 5 fine labels.

With mean substraction, training is performed on 32×32 images that have been shifted by up to 4 pixels in each direction with zero padding and random horizontal flipped. No other augmentation/deformation is used. We use ResNet-110 [20] as baseline network and generate the similarity for clustering by the prediction of baseline network. We set the number of cluster labels as 20 which is the same as the coarse labels of CIFAR-100.

Fig. 5 illustrates the cluster results, in which each black box represents a cluster label and each image represents a original fine label.

Based on the baseline network, AHLC add a new branch for 20 cluster labels with softmax and cross entropy loss, and by contrast, AHLM add a new branch for 20 coarse labels in the dataset by manually labeling. We test set λ_c as 1 for equal importance of two tasks, and set as 0.5 for promising the fine labels task dominant one. We also test the method in original data and augment data that have been shifted by up to 4 pixels in each direction with zero padding, mean subtraction and random horizontal flipped. Specifically, BaseLine is the original ResNet-110 with only 100 fine labels. AHLM is trained by the original 100 fine labels and 20 coarse labels with λ_c as 0.5 as well as AHLM-e with λ_c as 1. AHLC is

[2] http://www.cs.toronto.edu/ kriz/cifar.html

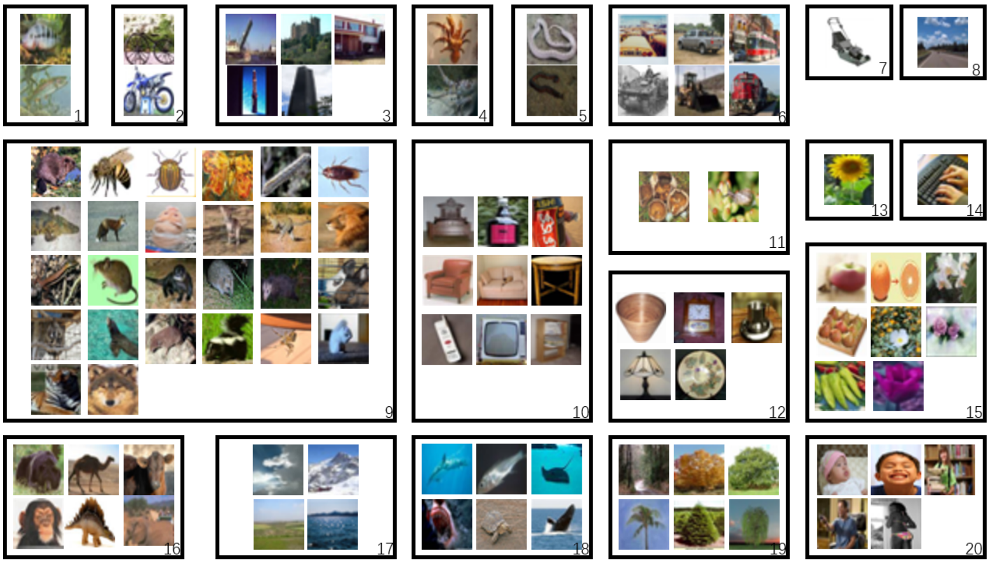

Fig. 5. Labels cluster result. Each black box represents a cluster label and each image represents an original fine label randomly chosen in dataset. The cluster(left to right, up to down) are: (1) aquarium fish, trout; (2) bicycle, motorcycle; (3) bridge, castle, house, rocket, skyscraper; (4) crab, lobster; (5) snake, worm; (6) bus, pickup truck, streetcar, tank, tractor, train; (7) lawn mower; (8) road; (9) beaver, bee, beetle, butterfly, caterpillar, cockroach, crocodile, fox, hamster, kangaroo, leopard, lion, lizard, mouse, otter, porcupine, possum, rabbit, raccoon, seal, shrew, skunk, spider, squirrel, tiger, wolf; (10) bed, bottle, can, chair, couch, table, telephone, television, wardrobe; (11) mushroom, snail; (12) bowl, clock, cup, lamp, plate; (13) sunflower; (14) keyboard; (15) apple, orange, orchid, pear, poppy, rose, sweet pepper, tulip; (16) bear, camel, cattle, chimpanzee, dinosaur, elephant; (17) cloud, mountain, plain, sea; (18) dolphin, flatfish, ray, shark, turtle, whale; (19) forest, maple tree, oak tree, palm tree, pine tree, willow tree; (20) baby, boy, girl, man, woman.

TABLE II

TEST ACCURACY ON CIFAR-100 DATASET. WITH MEAN SUBSTRACTION, TRAINING IS PERFORMED ON 32×32 IMAGES THAT HAVE BEEN SHIFTED BY UP TO 4 PIXELS IN EACH DIRECTION WITH ZERO PADDING AND RANDOM HORIZONTAL FLIPPED. NO OTHER AUGMENTATION/DEFORMATION IS USED. WE USE RESNET-110 [20] AS BASELINE NETWORK AND GENERATE THE SIMILARITY FOR CLUSTERING BY THE PREDICTION OF BASELINE NETWORK. WE SET THE NUMBER OF CLUSTER LABELS AS 20 WHICH IS THE SAME AS THE COARSE LABELS OF CIFAR-100. AHLM USES MANUAL COARSE LABELS.

Method	BaseLine	AHLM-e	AHLC-e	AHLM	**AHLC(proposed)**
Error without augment(%)	35.97	36.80	34.96	34.78	**34.23**
Error with augment(%)	26.90	27.39	26.99	26.64	**26.59**

trained by original 100 fine labels and 20 cluster labels with λ_c as 0.5 as well as AHLC-e with λ_c as 1.

Table II illustrates the test error of the original fine labels, which is tested on both raw data and augment data. AHLC gets the lowest test error in both raw data and augment data.

C. Result Analysis

For MNIST dataset, AHLC cluster the original 10 handwriting digits into 3 clusters. 1 and 7 in handwriting digits is easy to misclassify. 4 and 9 also share much features. AHLC cluster the similar labels into same cluster as expected. By adding prediction branch of cluster labels, it does improve the performance of the original classification task. The results indicates that we need make the target task a dominant one, otherwise, as Table I AHLC-e shows, the target task is slightly interfered by the adding task.

For CIFAR-100 dataset, AHLC cluster the original 100 fine labels into 20 labels. Compared with manual coarse labels, the cluster labels of AHLC are more likely to combine the similar label in visual together. As Fig. 5 shows, AHLC cluster bicycle, motorcycle into same cluster and bus, pickup truck, streetcar, tank, tractor, train into another cluster rather than manual coarse labels cluster bicycle, bus, motorcycle, pickup truck, train as vehicles 1 and cluster lawn-mower, rocket, streetcar, tank, tractor as vehicles 2, which bicycle and motorcycle are really different from the other vehicles in visual. AHLC also cluster forest, maple, oak, palm, pine, willow into same cluster and cloud, mountain, plain, sea into other cluster rather than manual coarse labels cluster forest into cluster large natural outdoor scenes consisted of cloud, mountain, plain, sea.

For classification performance, AHLC with cluster labels even reaches a better performance than AHLM with the manual coarse labels as Table II shows. For data without augment, which each label contains only 500 examples, AHLC and AHLM with hierarchical labels both perform a significant increase about 1.5% compared with baseline network, while for data with augment, they perform a slightly increase. It shows that the proposed AHLC do increase the classification performance especially for data with plenty of labels and few examples for each label.

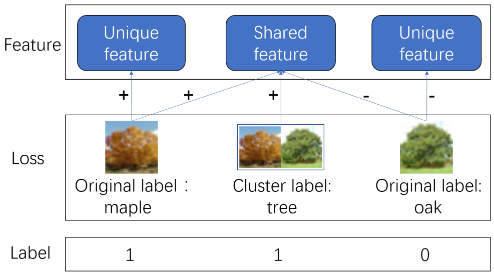

Fig. 6. Hierarchical labels help to learn hierarchical features. $+$ and $-$ mean the effect of enhancing and weakening.

Hierarchical labels help to learn hierarchical features. As Fig. 6 illustrates, an image with label maple or oak, it can only enhance unique features of maple or oak. But for the shared feature of maple and oak, at least one of the their labels is 0, so they can't enhance them simultaneously. By adding cluster label of tree, it will enhance shared features of maple and oak if either of them is 1. So hierarchical labels help to learn hierarchical features, which improve the performance of network.

V. Conclusion

In this paper, we proposed a method AHLC to improve the performance of classification with automatically hierarchical label clustering. The experimental results on the datasets of MNIST and CIFAR-100 show that: (a) AHLC clusters the similar labels efficiently; (b) Adding hierarchical labels improves classification performance of original task; (c) AHLC is a general method without extra information or architecture redesign.

Acknowledgment

This work was supported by National Natural Science Foundation of China (91520202, 81701785), Youth Innovation Promotion Association CAS, the CAS Scientific Research Equipment Development Project (YJKYYQ20170050) and the Beijing AI Project.

References

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 2014.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on International Conference on Machine Learning*, 2015, pp. 448–456.

[9] M. L. Zhang and Z. H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[10] L. Wang, "Improved multilabel classification with neural networks," in *International Conference on Parallel Problem Solving From Nature: PPSN X*, 2008, pp. 409–416.

[11] M. L. Zhang and Z. H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[12] Y. Zhu, K. M. Ting, and Z. H. Zhou, "Multi-label learning with emerging new labels," in *IEEE International Conference on Data Mining*, 2017, pp. 1371–1376.

[13] J. R. Quinlan, "Induction on decision tree," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[14] J. F. Zhi-Hua Zhou, "Deep forest: Towards an alternative to deep neural networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3553–3559.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[16] K. Zhao, W. S. Chu, l. T. F. De, J. F. Cohn, and H. Zhang, "Joint patch and multi-label learning for facial action unit and holistic expression recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3931–3946, 2016.

[17] K. Zhao, W. S. Chu, and H. Zhang, "Deep region and multi-label learning for facial action unit detection," in *Computer Vision and Pattern Recognition*, 2016, pp. 3391–3399.

[18] S. C. Johnson, "Hierarchical clustering schemes." *Psychometrika*, vol. 32, no. 3, p. 241, 1967.

[19] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3859–3869.

[20] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Computer Vision and Pattern Recognition*, pp. 2261–2269, 2016.