International Journal of Approximate Reasoning 149 (2022) 61–69

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Hierarchical classification based on coarse- to fine-grained knowledge transfer

Zeyu Qiu [a,b], Minjie Hu [a,b], Hong Zhao [a,b,*]

[a] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China
[b] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian, 363000, China

A R T I C L E I N F O

Article history:
Received 7 November 2021
Received in revised form 5 July 2022
Accepted 6 July 2022
Available online 13 July 2022

Keywords:
Hierarchical classification
Deep convolutional network
Branch network
Class hierarchical structure

A B S T R A C T

Hierarchical classification has become one of the most popular research topics because the scale of data has increased exponentially. Top-down hierarchical classification is an effective classification method using the hierarchical class structure as important side information. However, inter-level error propagation is a crucial problem in the top-down strategy. In this paper, we propose a hierarchical classifier based on deep branch convolutional neural networks, which achieves hierarchical classification based on coarse- to fine-grained knowledge transfer. Specifically, we use a deep convolutional neural network to extract image rough and detailed features from shallow and deep networks. We then embed the branch network at different depths of the convolutional network for hierarchical classification. We splice features from the previous branch network to the current branch. It alleviates inter-level error propagation by knowledge transfer from coarse- to fine-grained. Experimental results on four datasets show that the proposed method outperforms nine popular classifiers for hierarchical classification. Especially on the CIFAR10 dataset, our method is about 5% better than the second-best method.

1. Introduction

With the advent of the big data era, the scale of data has increased exponentially, and numerous classes have become one of the main challenges of classification technology [38]. Fortunately, there is always a hierarchical structure among numerous classes, which are usually represented in a class hierarchical tree [22,33]. The hierarchical structure is generally constructed by the semantic subordination among the coarse- and fine-grained classes in WordNet [21]. As prior knowledge, it is one of the important auxiliary information for image classification. In recent years, growing attention has been paid to structured or hierarchical classification learning [32,36].

The hierarchical class structure provides important side information for classification [38]. There are close relationships between classes in the class hierarchy structure. The upper-level class is coarse-grained relative to the lower level, and the lower-level class is fine-grained relative to the upper level. An example of a hierarchical class tree is shown in Fig. 1. *Elephant* belongs to the *Terrestrial* class in the fine-grained and it also belongs to the *Animal* class in the coarse-grained. According to the class semantic hierarchy characters of the data, the model can easily distinguish the coarse-grained classes during the classification process, which guides the model to discriminate the fine-grained classes further.

* Corresponding author at: School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China.
E-mail address: hongzhaocn@163.com (H. Zhao).

https://doi.org/10.1016/j.ijar.2022.07.002
0888-613X/

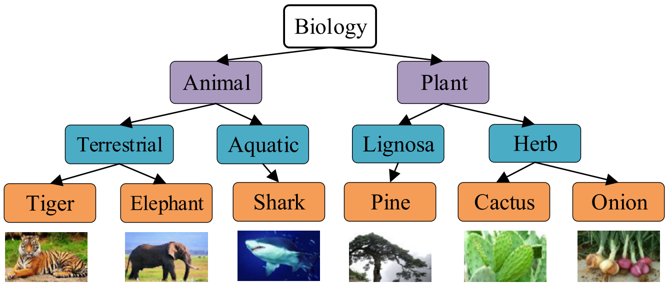

Fig. 1. An example of class hierarchical tree.

Hierarchical classification is a particular classification task in machine learning and has been widely studied [13,19,39]. There are many deep researches on constructing hierarchical class trees for hierarchical classification [11,17,23]. Liu et al. [18] developed a method called recursive nonnegative matrix decomposition, which uses a linear classifier to propagate the data instances on the internal nodes of the tree to their corresponding leaf nodes. Mohan et al. [22] discovered critical sections consisting of nodes in the tree structure are interdependent due to the ancestor-successor relationship, thus achieving higher concurrency and better performance in a multithreaded system.

Top-down hierarchical classification is one of the most effective classification methods [6,14,28]. It usually classifies an object level by level into a real class in the class hierarchical tree [31,34]. For example, Korinna et al. [1] found that users tend to browse the hierarchy in a top-down manner when searching for documents in the topic hierarchy and propose a user-oriented hierarchical classification method based on the decision theory framework. Also, Bewley et al. [3] used binary classifiers with probabilistic output and applied "one-vs-rest" classification at each hierarchy level for prediction. Especially, Cevikalp [5] divided each node of a group of classes in the hierarchy into two smaller groups from top to bottom, greatly shortening the test time for the clustering algorithm.

Unlike previous classification methods, which use a single branch CNN network, some researchers proposed a multi-branch network to address hierarchical classification. For instance, Zhu et al. [40] proposed a hierarchical classification method to establish several branch networks in the neural network, improving classification accuracy. Similarly, Lam et al. [15] performed coarse- and fine-grained classification in the group, then used triplet loss training to play the role of a classifier in the classification hierarchy. The approaches mentioned above use top-down strategies to tackle the challenges of hierarchical classification. They have achieved effective results because they take full advantage of class dependencies. However, the top-down strategy has a serious inter-level error propagation problem. With the increase of the depth of the hierarchical classification tree, the misclassification in the upper level will gradually accumulate and pass to the lower level, making the classification result in the lower level far deviates from the real class.

In this paper, we propose a hierarchical classification based on coarse- to fine-grained knowledge transfer, which simultaneously uses the rough and detailed features of different layers to alleviate inter-level error propagation. It achieves the goal of hierarchical classification using the currently popular neural network as classification. Specifically, the shallow layer of the convolutional network extracts some rough features of the image, such as the contour of an object. Therefore, the branches embedded in the shallow network are used to predict coarse-grained classes. The deep network can capture more detailed features [30], so the branches embedded in the deep network are used to predict the finer-grained classes or true classes. We also consider that the coarse-grained predictions can help fine-grained because the coarse-grained classes are the parent class of the fine-grained classes. We re-embed the branches from the shallower layer into the deeper network as a knowledge transfer channel to transfer knowledge from coarse- to fine-grained classification tasks. Unlike traditional hierarchical classification, the upper and lower levels can automatically adjust the classification results through the neural network parameters. Additionally, the knowledge transfer channel established between levels can further maintain the consistency of the classification results of the upper and lower levels. It alleviates the problem of error propagation between levels.

We verify the effectiveness of the proposed method on four datasets with a class hierarchical structure. For the ablation experiment, we discuss and confirm the role of each part of the model. We control the experimental variables and compare the model with nine popular neural network models. The experimental results show that our model performs better than the comparison model on the four datasets.

The rest of the paper is arranged as follows. Section 2 elaborates on the model structure and loss function of our proposed model. Sections 3 and 4 show the experimental setting, results and analysis. Finally, Section 5 summarizes the paper and discusses future work.

2. Deep branch convolutional neural network

This section describes the details of our model. The overall framework of the model is given in Section 2.1, the backbone network of the model is introduced in Section 2.2, the branch network and splicing strategy are shown in Section 2.3, and the expression of the loss function is listed in Section 2.4.

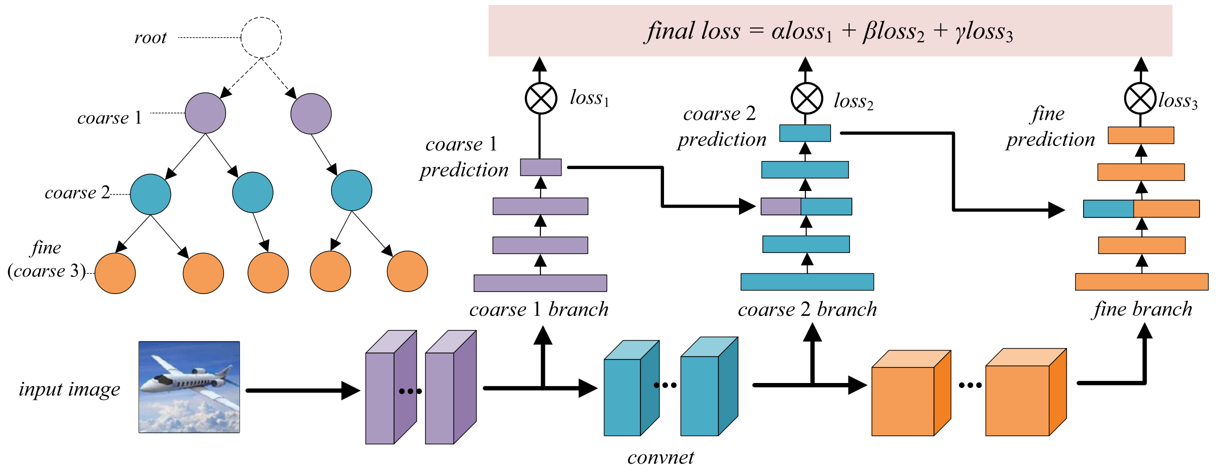

Fig. 2. Branch deep convolutional neural network.

2.1. Model overview

Hierarchical classification and neural networks are both relatively mature research directions. Many studies have been conducted, but few have used neural networks for hierarchical classification. In this paper, we design a deep branch convolutional neural network model for hierarchical classification (HBCNN), as shown in Fig. 2. The model consists of three parts: class hierarchical tree of a dataset, backbone network, and branch network. Each node in the hierarchical tree represents a class. The model uses the hierarchical tree (top left) of the dataset to gradually classify one input image into the final class through hierarchical classification (bottom right). The specific classification process is described below.

First, some features of the input image are roughly extracted from the shallow convolution kernel of the backbone network and then transferred to the first branch network to classify the image into the first level class (*coarse* 1) in the hierarchical tree.

Then, after the input image is further extracted features through the middle convolution kernel, it works with the prediction from the first branch to classify the image into the second level class (*coarse* 2) in the hierarchical tree.

Finally, the prediction result of the second branch helps the depth features extracted by the deep convolution kernel to classify the image into the real ($fine$) class.

2.2. Backbone network

The convolutional neural network is an efficient recognition method developed in recent years. Features with different levels can be extracted from the image by setting the size and number of convolution kernels. The convolutional network structure includes two groups of layers. One is a convolutional layer, which is used to extract image features. The other is a fully connected layer, which maps features extracted from the convolution layer to the target class [25].

The HBCNN for image classification can be represented by $c_k = M_k(c_{k-1}, E_k(x))$, where x is an input image, E_k is a feature extraction function for predicting the k^{th} level class, M_k is a feature mapping function for predicting the k^{th} level class, and c_k is the k^{th} level prediction class of the image. Like most neural network models, our model uses convolutional and fully connected networks to complete the feature extraction and mapping process. The convolutional network is used for feature extraction as the backbone network, and the fully connected network is used for feature mapping as the branch network. In this section, we mainly introduce the architecture of the backbone network.

We use the convolutional part of the VGG16 model as the backbone network of the model. Its architecture is listed in Table 1, where *image* is the input image, and $(conv3-64) \times 2$ indicates that the two convolutional layers use 64 convolution kernels with a size of 3×3. Parameter $maxpool-2$ means that a 2×2 size window is used for maximum pooling. Also, branches from *corase* 1 and *corase* 2 indicate a fully connected branch mapped to the first and second level classes after the convolutional layer. Finally, the $fine$ branch means that the last branch is mapped to the lowest class in the hierarchical tree, and the final prediction result is obtained.

The VGG16 model is composed of several convolution layers and pool layers stacked. The number of convolution cores used increases from shallow layer to deep layer. It aligns with the idea of extracting features with different degrees of fineness from coarse to fine in hierarchical classification. Moreover, the VGG model has a variety of network structures to choose from and has strong scalability, which contributes to building more branch networks.

2.3. Branch network and splicing strategy

The branch network is mainly used to map the features of different fineness extracted by the convolutional neural network to the class of the corresponding level in the hierarchical tree. We use the prediction of the previous fully connected

Table 1
The architecture of the backbone network.

$image$	
$(conv3-64)\times 2$	
$maxpool-2$	
$(conv3-128)\times 2$	
$maxpool-2$	
$\downarrow\ \rightarrow$	$coarse\ 1\ branch$
$(conv3-256)\times 3$	
$maxpool-2$	
$\downarrow\ \rightarrow$	$coarse\ 2\ branch$
$(conv3-256)\times 3$	
$(conv3-256)\times 3$	
$\rightarrow$	$fine\ branch$

Table 2
The architecture of the branch network.

$coarse\ 1\ branch$	$coarse\ 2\ branch$	$fine\ branch$
$flatten$	$flatten$	$flatten$
$fc-256$	$fc-n_2$	$fc-n_3$
$fc-256$	$*fc-n_1\oplus *fc-n_2$	$*fc-n_2\oplus *fc-n_3$
$fc-n_1$	$fc-1024$	$fc-4096$
	$fc-n_2$	$fc-n_f$

branch to help predict the next branch. In a class hierarchy tree, classes are always predicted level by level from top to bottom. The correct prediction of the coarse-grained classes can promote the correct prediction of the fine-grained classes. We adopt a splicing strategy, which uses the output result of the previous branch as a feature to splice with the fully connected layer of the next branch to limit the prediction range of the next branch. It is consistent with human classification thinking.

The structure of the three fully connected branches and the implementation method of the splicing strategy are listed in Table 2. Among them, $flatten$ indicates that the high-dimensional feature vector from the convolutional neural network is straightened into a one-dimensional vector as the input of the fully connected network. $fc-256$ means that a fully connected layer composed of 256 neurons is used. $fc-n_1$ represents a fully connected layer, and the number of using neurons is the number of classes in the first level of the hierarchical tree. While $*fc-n_1$ means that the outputs of $fc-n_1$ from $coarse$ 1 $branch$ and $fc-n_2$ means that the outputs of $fc-n_2$ from $coarse$ 2 $branch$. $*fc-n_1\oplus *fc-n_2$ means that the outputs of $fc-n_1$ and $fc-n_2$ are horizontally spliced into a one-dimensional vector in $coarse$ 2 $branch$, and then input to the next level. $fc-n_f$ represents a fully connected layer composed of neurons with the number of $fine$ classes, which is the number of classes at the bottom of the hierarchical tree.

The construction of a branch network realizes a single sample classification at different granularity. Unlike traditional single classification, it has information transmission and is no longer an independent task. The splicing strategy makes the classification results on coarse granularity promote the correct classification on fine granularity and alleviate the error propagation. This makes it possible for large-scale data such as ImageNet to realize accurate hierarchical classification on neural networks.

2.4. Loss function

In our model, we leverage three-branch networks for prediction, and the three-branch networks affect each other. Therefore, we synchronously establish a fusion loss and consider the losses of the three branches to optimize the model. Different branches are given different weights. In addition, we also regularize the parameters of the three-branch networks to avoid the overfitting problem. Let $(x_i, y_i^k)\in I$ be an image sample, where I is sample set, x_i is a feature matrix of the i^{th} sample, and y_i^k is the true label of x_i on the k^{th} level of the hierarchical tree. The predicted probability of x_i is y_i^k as: $p^k = F_{w_k}(x_i)|x\in y_i^k$, where w_k is a parameter of the k^{th} network branch. Then the cross-entropy loss function of HBCNN can be defined as:

$$L_i = \sum_k -A_k log\ \{F_{w_k}(x_i)|x_i\in y_i^k\}, \tag{1}$$

where $A_k=\{\alpha,\beta,\gamma\}$ is the loss weight of three branches. We use an l_2 regularization term [10] in our model. Therefore, the total loss can be written as:

$$L_i = \sum_k -A_k log\ \{F_{w_k}(x_i)|x_i\in y_i^k\} + \lambda||w_k||^2, \tag{2}$$

Table 3
Dataset description.

Name	Train	Test	Size	n_1	n_2	n_f
CIFAR10	50,000	10,000	$32 \times 32 \times 3$	2	7	10
CIFAR100	50,000	10,000	$32 \times 32 \times 3$	8	20	100
ILSVRC	12,346	11,845	$64 \times 64 \times 1$	2	5	57
VOC	4,883	2,093	$25 \times 40 \times 1$	1	4	20

where $||w_k||^2$ is the penalty term and λ is the weight of penalty term.

The loss function integrates the loss of each branch, and the setting of the loss weight A_k balances the optimization speed of each branch so that the classification task at each level achieves a good result. The addition penalty term l_2 weakens the model sensitivity for the training data, which prevents overfitting [26].

3. Experiment setting

All experiments are implemented on a computer with four 2.27 GHz processors and 12 GB of available GPU. Datasets used in the experiment are described in Section 3.1. Comparison methods are introduced in Section 3.2.

3.1. Dataset

We use four image datasets — CIFAR10, CIFAR100, ILSVRC, and VOC — with the class hierarchical structure in the experiment. Table 3 lists specific information of these datasets, where n_1, n_2, and n_f represent the number of classes of the dataset in *coarse* 1, *coarse* 2 and *fine* granularity, respectively. The details of them are introduced as follows.

(1) The CIFAR10 dataset comprises 60,000 32×32 color images in 10 classes, and each class has 6,000 images. The original dataset has no class hierarchy. We use the method in the paper [40] to manually construct a three-level class hierarchical tree based on the semantics of the classes [16].

(2) The CIFAR100 dataset consists of 60,000 32×32 color images in 100 classes, and each class contains 600 images. The 100 classes in the dataset are divided into 20 superclasses. Each image has a “fine” label and a “rough” label. Based on the rough label, we use the method of the paper [40] to manually construct the upper-level class according to the semantics of class [2].

(3) The ILSVRC dataset is a subset of ImageNet consisting of 24,194 64×64 grayscale images in 57 classes. The dataset originally had a three-level class hierarchy with 2, 5, and 57 classes on the three levels, respectively [20].

(4) The VOC is a standardized dataset for detection and recognition. We use 6,976 grayscale images of 25×40 size after preprocessing. The dataset originally had a class hierarchy [12].

3.2. Comparison methods

In this section, we compare the proposed model with several methods. Their details are as follows:

- MnistNet is a simple neural network composed of 3 groups of 3×3 and 1×1 convolution kernels and three fully connected layers [29].
- TSL16 is a 16-layer neural network composed of 13 convolutional layers and 3 fully connected layers [8].
- LeNet was proposed by Professor Yann LeCun. It is the first convolutional neural network successfully applied to digital recognition problems [4].
- AlexNet won the championship in the ImageNet image classification task competition in 2012. It is a blockbuster and created an unprecedented climax for deep neural networks [27].
- ZFNet is an improvement of AlexNet. It prevents the network from focusing only on the high-frequency and low-frequency characteristics of the image while ignoring the mid-frequency characteristics [37].
- Resnet18 is a deep residual network. Its basic idea is to introduce a “shortcut connection” that can skip one or more layers. ResNet18 consists of 18 network layers, while ResNet34 contains 34 layers [35,9].
- VGG refreshes the performance by continuously deepening the network structure and increasing the network depth to ensure learning of more complex patterns. VGG16 and VGG19 have 16 and 19 network layers, respectively [24,7].

Since the part of our model uses VGG16, the performance can be better reflected by comparing it with VGG16 and VGG19. In addition, LeNet, AlexNet, and ResNet18 are currently popular models that have achieved great results in the image field. The complexity of the network structure is comparable to our model.

Table 4
Accuracy of different granularity on four datasets (%).

Branch	CIFAR10	CIFAR100	ILSVRC	VOC
$coarse$ 1	97.45	77.56	99.83	100
$coarse$ 2	91.88	75.37	98.62	94.60
$fine$	89.33	66.23	86.93	94.31

Table 5
The results of ablation experiment (%).

Name	CIFAR10	CIFAR100	ILSVRC	VOC
$baseline$	83.25	62.61	84.19	90.35
$baseline + branch$	84.15	63.91	83.90	90.11
$baseline + branch + splicing$	83.92	64.21	85.66	**94.65**
$baseline + branch + l_2$	88.52	65.40	85.68	90.01
$baseline + branch + splicing + l_2$	**89.33**	**66.23**	**86.93**	94.31

4. Experimental results and analysis

In this section, we verify the performance of the model through two set experiments. Section 4.1 shows the hierarchical classification results of our model. The results of the ablation experiment are shown in Section 4.2. Then comparison with the latest image classification method is listed in Section 4.3.

4.1. The result of hierarchical classification

In the experiment, the training epoch is 120, and the weight A_k is $\{0.2, 0.3, 0.4\}$ for all datasets. The loss weights of $corase$ 1 and $coarse$ 2 branches are 0.2 and 0.3, respectively. The loss weight of the last branch is 0.4. Balancing the branches makes the model pay more attention to the final classification result. Table 4 shows the results of hierarchical classification from three branches of our model.

From Table 4, we can obtain the following:

(1) The classification accuracy on $coarse$ 1 granularity is the highest, and the accuracy on $coarse$ 2 and $fine$ granularity decreases. This is because $coarse$ 1 granularity has the least classes. In contrast, $fine$ granularity has the most classes, and the upper-level hierarchical classification errors are easily transmitted to the lower level.

(2) The classification accuracy is as high as 100% in the $coarse$ 1 granularity of VOC because it has only one class in the $coarse$ 1 granularity. Our model can adapt to multi-granular class hierarchy datasets rather than only working at three granularities.

4.2. Ablation experiment

All settings in the ablation experiment are consistent with the previous section. Then we use the classification accuracy of the VGG16 network as the baseline. Based on the baseline, we add the elements of our model in turn and conduct experiments on the above four datasets to record the classification accuracy of the $fine$ branch to verify the effectiveness of our model. Table 5 lists the results of ablation experiments. The number in bold represents the best result on the column dataset.

From Table 5, we can obtain the following conclusions:

(1) $baseline + branch$: After adding three-branch networks to the baseline, the accuracy of the CIFAR10, CIFAR100, and VOC datasets has been improved. There is a slight decrease from 84.19 to 83.90 on the ILSVRC dataset. This is due to the high feature dimension of the ILSVRC dataset, which requires higher training intensity. However, the existence of branches distributes the training intensity of the model so that the model does not achieve the best effect.

(2) $baseline + branch + splicing$: Compared with the baseline, the accuracy of the four datasets has improved. Especially on the VOC dataset, the accuracy is improved by 4.3%. Compared with the result in $baseline + branch$, the classification accuracy of the other three datasets is improved except for CIFAR10. This is because the hierarchical class tree of CIFAR10 is manually constructed, and it is not entirely reasonable for some classes of aggregates.

(3) $baseline + branch + l_2$: Compared with the baseline, the accuracy of the model on the VOC dataset has decreased slightly, and the accuracy of the other three datasets has been greatly improved. Compared with the result in $baseline + branch$, there is a similar situation because it does not have many changes when the branch network and l_2 regularization are added.

(4) $baseline + branch + splicing + l_2$: The accuracy of the model on the four datasets has been greatly improved. Compared with $baseline$, it has improved by about 5% on CIFAR10 and about 4% on CIFAR100. ILSVRC and VOC improve about 1% and 5%, respectively. Due to the relatively fewer categories of CIFAR10 and VOC, a more significant improvement is achieved under the same setting. Hierarchical classification on VOC datasets with 1 class at $coarse$ 1 granularity is equivalent to only two granularities of $coarse$ 2 and $fine$, so the inter-level error propagation is slight. In summary, the effectiveness of our model can be proved.

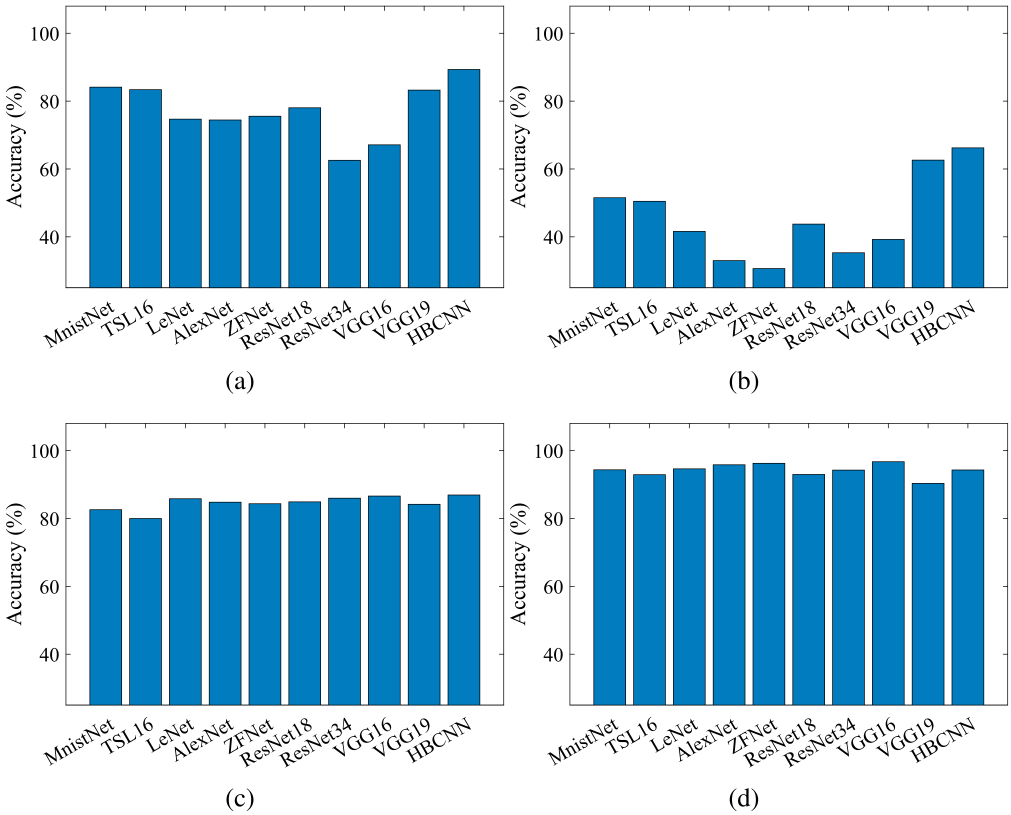

Fig. 3. Accuracy of the different methods on four datasets. (a) CIFAR10; (b) CIFAR100; (c) ILSVRC; and (d) VOC.

4.3. Comparative experiment

This section reports the comparison results of our model with other advanced models. For the fairness of the experiment, we keep all the super parameters consistent and set the epoch to 120. Since the comparison methods do not have the function of hierarchical classification, we only compare the accuracy of the final class. Fig. 3 shows the results of the comparative experiment.

From Fig. 3, we can obtain the following observations:

(1) On the CIFAR10 and CIFAR100 datasets, our model has great advantages over the compared networks, which are shown in Figs. 3 (a) and (b), respectively. AlexNet and ZFNet cause poor classification accuracy due to their simple network structure. Although MnistNet, TSL16, and VGG16 are better, they are still not as well as HBCNN, which are 5.21%, 5.45% and 6.08% lower than our model, respectively. ResNet18, ResNet34, and VGG19 have similar and deeper network structures to VGG16, but the classification accuracy is not as good as VGG16. Compared with the deeper network, our model has its advantage because it constructs three branches to balance the difficulty of the classification task and the complexity of the network.

(2) On the ILSVRC dataset, the effects of each algorithm are very close, but our model is still slightly better than other models. As shown in Fig. 3 (c), compared with ZFNet, which ranks second in accuracy, and the accuracy of our model is 0.39% better than it. Compared with the TSL16 model, the accuracy of our model is 6.93% better than it.

(3) As shown in Fig. 3 (d), on the VOC dataset, although the classification effect of ZFNet, ResNet18, and ResNet34 is slightly better than our model, which is 2.43%, 1.53%, and 1.96% higher than our model, respectively. The VOC dataset follows the long-tailed distribution, which is imbalanced data distribution. This skewed long-tailed data distribution is the main reason for the poor effect of our model.

In our experiment, the classification results of our model at different granularities are shown to confirm its special functions. Ablation experiments on four datasets verify the necessity of each component. In addition, the splicing strategy alleviates the problem of inter-level error propagation and effectively improves the accuracy of classification.

5. Conclusions and future work

This paper proposes a hierarchical classification method based on deep branch convolutional neural networks. We embed the fully connected branch networks at different depths of the convolutional network and construct classification tasks with different granularities on each branch. Then we leverage the splicing strategy to make the classification results of coarse-grained tasks help classify fine-grained tasks by knowledge transfer and use a loss function with penalty terms to prevent overfitting. Finally, the effectiveness of the model is verified by ablation experiments and comparative experiments on four datasets. In the future, we will study whether the branch network has the function of preventing overfitting and how to eliminate the problem of gradient disappearance when there are too many branches.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62141602 and the Natural Science Foundation of Fujian Province under Grant Nos. 2021J011003 and 2021J011006.

References

[1] K. Bade, E. Hullermeier, A. Nurnberger, Hierarchical classification by expected utility maximization, in: International Conference on Data Mining, 2006, pp. 43–52.
[2] B. Barz, J. Denzler, Do we train on test data? Purging CIFAR of near-duplicates, J. Imaging 6 (6) (2020) 41–52.
[3] S.M. Bewley, N. Nourani-Vatani, D. Rao, Hierarchical classification in auv imagery, in: Springer Tracts in Advanced Robotics, 2015, pp. 3–16.
[4] A. Bouti, M.A. Mahraz, J. Riffi, A robust system for road sign detection and classification using lenet architecture based on convolutional neural network, Soft Comput. 24 (9) (2020) 1–12.
[5] H. Cevikalp, New clustering algorithms for the support vector machine based hierarchical classification, Pattern Recognit. Lett. 31 (11) (2010) 1285–1291.
[6] N. Cheng, Y. Zou, Y. Teng, On the method of personal attributes extraction based on textual knowledge and hierarchical classification, Appl. Linguist. (2019).
[7] N. Dey, Y.D. Zhang, V. Rajinikanth, Customized VGG19 architecture for pneumonia detection in chest x-rays, Pattern Recognit. Lett. 143 (10) (2021) 67–74.
[8] E. Saykol, H.T. Türe, A.M. Sirvanci, M. Turan, Posture labeling based gesture classification for Turkish sign language using depth values, Kybernetes 45 (4) (2016) 604–621.
[9] Y. Fu, C. Aldrich, Flotation froth image recognition with convolutional neural networks, Miner. Eng. 132 (2019) 183–190.
[10] J. Gou, B. Hou, Y. Yuan, A new discriminative collaborative representation-based classification method via l2 regularizations, Neural Comput. Appl. 32 (8) (2020) 9479–9493.
[11] F. Harary, E. Palmer, R. Read, The number of ways to label a structure, Psychometrika 32 (2) (1967) 155–156.
[12] T. Hu, P. Mettes, J.H. Huang, Silco: show a few images, localize the common object, in: International Conference on Computer Vision, 2019, pp. 35–47.
[13] C.N. Silla Jr., A.A. Freitas, A survey of hierarchical classification across different application domains, Data Min. Knowl. Discov. (2011) 31–72.
[14] K. Sendin, M. Manley, F. Marini, P.J. Williams, Hierarchical classification pathway for white maize, defect and foreign material classification using spectral imaging, Microchem. J. 162 (10) (2020) 1–9.
[15] P. Lam, M. Ian, P. Huy, Deep feature embedding and hierarchical classification for audio scene classification, in: International Joint Conference on Neural Networks, 2020, pp. 1–7.
[16] H. Li, H. Liu, X. Ji, Cifar10-dvs: an event-stream dataset for object classification, Front. Neurosci. 11 (2017) 41–55.
[17] G. Lin, C. Fan, W. Chen, Class label autoencoder with structure refinement for zero-shot learning, Neurocomputing 428 (2020).
[18] L. Liu, P.M. Comar, S. Saha, Recursive NMF: efficient label tree learning for large multi-class problems, in: Pattern Recognition, 2012.
[19] Y. Liu, Y. Dou, R. Jin, Hierarchical learning with backtracking algorithm based on the visual confusion label tree for large-scale image classification, Vis. Comput. 4 (10) (2021).
[20] C. Lu, H. Chen, Q. Chen, 1-HKUST: object detection in ILSVRC 2014, Comput. Sci. 42 (6) (2014) 837–845.
[21] G.A. Miller, Wordnet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[22] A. Mohan, G. Singhal, B. Biswas, A lock management framework for a class hierarchy tree, in: Advances in Intelligent Systems and Computing, vol. 174, 2013, pp. 297–302.
[23] M. Mostajabi, M. Maire, G. Shakhnarovich, Regularizing deep networks by modeling and predicting label structure, in: Computer Vision and Pattern Recognition, 2018.
[24] A.K. Rangarajan, R. Purushothaman, Disease classification in eggplant using pre-trained VGG16 and msvm, Sci. Rep. 10 (1) (2020) 23–42.
[25] A. Ren, J. Li, Z. Li, SC-DCNN: highly-scalable deep convolutional neural network using stochastic computing, Oper. Syst. Rev. 51 (2) (2016) 405–418.
[26] C. Sabett, A. Hafftka, K. Sexton, L1, lp, l2, and elastic net penalties for regularization of gaussian component distributions in magnetic resonance relaxometry, Concepts Magn. Reson., Part A 46 (2) (2017) 303–315.
[27] S. Sadhana, R. Mallika, An intelligent technique for detection of diabetic retinopathy using improved alexnet model based convoluitonal neural network, J. Intell. Fuzzy Syst. 40 (1) (2020) 1–12.
[28] W. Shen, Z. Wei, Q. Li, Three-way decisions based blocking reduction models in hierarchical classification, Inf. Sci. 523 (4) (2020).
[29] S. Tabik, R.F. Alvear-Sandoval, Mnist-net10: a heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate, Inf. Fusion 62 (2020) 73–80.
[30] Y. Wang, R. Zou, F. Liu, L. Zhang, Q. Liu, A review of wind speed and wind power forecasting with deep neural networks, Appl. Energy (2021).
[31] C. Wen, W. Yang, J. Tang, Community question topic categorization via hierarchical kernelized classification, in: Conference on Information and Knowledge Management, 2013.
[32] Q.Y. Wu, M.K. Tan, H.J. Song, Ml-forest: a multi-label tree ensemble method for multi-label classification, IEEE Trans. Knowl. Data Eng. 28 (10) (2016) 2665–2680.
[33] J. Xiao, Y.E. Feng, Research on data mining algorithm based on conception hierarchy tree and its application, Appl. Res. Comput. (2005).
[34] D.U. Yongping, D.U. Xiaoyan, S. Chen, Relaxed Hierarchy Structure Construction for Text Classification, Beijing University of Technology, 2017.
[35] X. Yu, S.H. Wang, Abnormality diagnosis in mammograms by transfer learning based on resnet18, Fundam. Inform. 168 (2) (2019) 219–230.

[36] Z.Q. Yuan, C.S. Xu, J.T. Sang, Learning feature hierarchies: a layer-wise tag-embedded approach, IEEE Trans. Multimed. 17 (6) (2015) 816–827.
[37] D.M. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818–833.
[38] H. Zhao, Q. Hu, P. Zhu, A recursive regularization based feature selection framework for hierarchical classification, IEEE Trans. Knowl. Data Eng. 33 (7) (2021) 2833–2846.
[39] H. Zhao, P. Wang, Q. Hu, Fuzzy rough set based feature selection for large-scale hierarchical classification, IEEE Trans. Fuzzy Syst. 27 (10) (2019) 1891–1903.
[40] X. Zhu, M. Bain, B-CNN: branch convolutional neural network for hierarchical classification, arXiv: Computer Vision and Pattern Recognition, 2017.