Hierarchical bilinear convolutional neural network for image classification

Xiang Zhang | Lei Tang | Hangzai Luo | Sheng Zhong | Ziyu Guan | Jinye Peng | Jianping Fan

1School of Information and Technology, Northwest University, Shaanxi, 710127, China
2Center of Innovations, Xi’an Microelectronics Technology Institute, Shaanxi, China
3School of Communications and Information Engineering, Xi’an University of Posts and Telecommunications, Shaanxi, 710061, China
4Department of Computer Science, UNC, Charlotte, NC, 28223, USA

Abstract
Image classification is one of the mainstream tasks of computer vision. However, most existing methods use labels of the same granularity level for training. This leads to ignoring the hierarchy that may help to differentiate different visual objects better. Embedding hierarchical information into the convolutional neural networks (CNNs) can effectively regulate the semantic space and thus reduce the ambiguity of prediction. To this end, a multi-task learning framework, named as Hierarchical Bilinear Convolutional Neural Network (HB-CNN), is developed by seamlessly integrating CNNs with multi-task learning over the hierarchical visual concept structures. Specifically, the labels with a hierarchical structure are used as the supervision to hierarchically train multiple branch networks. In this way, the model can not only learn additional information (e.g. context information) as the coarse-level category features, but also focus the learned fine-level category features on the object properties. To smoothly pass hierarchical conceptual information and encourage feature reuse, a connectivity pattern is proposed to connect features at different levels. Furthermore, a bilinear module is embedded to generalise various orderless texture feature descriptors so that our model can capture more discriminative features. The proposed method is extensively evaluated on the CIFAR-10, CIFAR-100, and ‘Orchid’ Plant image sets. The experimental results show the effectiveness and superiority of our method.

1 | INTRODUCTION

Convolutional Neural Networks (CNNs) have demonstrated their outstanding capability on the task of image classification and achieved impressive performances [1–5]. However, most state-of-the-art methods have a common defect: the training labels lose the hierarchical structure information of categories since they are at the same level of granularity. Ignoring the hierarchical semantic relations implied in an object would result in losing useful discriminative information for the classification task.

Psychologists suggest that the human visual system has a hierarchical structure, and it already has the basic-level visual classification (e.g. dog) ability before fine visual classification (e.g. pomeranian) [6]. A typical hierarchical classification scenario is: when identifying a snake, humans commonly follow a coarse-to-fine flow 'animal → serpentes → snake'. One big advantage of hierarchical classification is that the error can be restricted to a subcategory. This hierarchy improves classification performance for three reasons: (1) This hierarchical structure encodes rich context information across different levels. For instance, a calamondin orange may not be easy to differentiate from a yellow ping pong ball. As we all know, the yellow ping pong ball belongs to sports equipment and calamondin orange belongs to fruits. Therefore, it is easy to tell them apart because the environment (context information) between the fruit and the sports equipment is so different. (2) This hierarchy can capture multi-granularity semantic concepts, thus enriching the object features. For instance, a nereid may not be easy to differentiate from a centipeda. In biology, nereid (fine-level category label) belongs to the annelid (coarse-level category label) and centipede (fine-level category label) belongs to the arthropoda (coarse-level category label). The biggest difference between the annelid and the arthropoda is that the
The human visual system is homonomous metamerism (each segment of an animal is basically the same) and the arthropoda is heteronomous metamerism (each segment of an animal is distinctly different). Specific distinguishing features (homonomous metamerism vs. heteronomous metamerism) that are ignored by the fine-level category label (nereid vs. centipeda) can be captured from the coarse-level category label (annelid vs. arthropoda). (3) This hierarchical classification is a multi-task learning method, which can effectively learn coarse-to-fine feature representation by jointly optimising coarse category classifier and fine category classifier. In order to better simulate the human visual system, machines should have the ability to attend more specific information at different levels of granularity. An intuitive alternative is to organise the network structure in a hierarchical manner according to the divide-and-conquer strategy.

Herein, a multi-task learning framework, named as Hierarchical Bilinear Convolutional Neural Network (HB-CNN), is developed by seamlessly integrating CNNs with multi-task learning over the hierarchical visual concept structures. Specifically, it contains multiple branch networks along the main convolution workflow (e.g. VGG16 [2]) to do predictions hierarchically. In this way, the model can learn coarse to fine concepts at the output stage. To smoothly pass hierarchical conceptual information and encourage feature reuse, a connectivity pattern is proposed to connect the features of multiple output branches. Furthermore, we embed the bilinear module [7] into this special CNN to better model subtle visual differences, thereby improving classification performance. Besides, inspired by the work in Ref. [8], we use a hierarchical training strategy that can enhance the model performance and prevent the impact of vanishing gradient problem. This training strategy enables lower-level parameters to be activated and trained earlier than higher-level parameters. Our results on CIFAR-10, CIFAR-100, and ‘Orchid’ Plant image sets suggest that the proposed method outperforms the corresponding CNN-based baselines.

In Section 2, We briefly review the related works of image classification. Then we describe the proposed HB-CNN method in Section 3. The experimental results and analysis are shown in Section 4. Last, We summarise this work in Section 5.

2 RELATED WORK

Convolutional neural networks (CNNs) have recently achieved great success in image recognition. Recently, many methods were proposed to improve classification performance. For example, a bilinear model [7] was proposed to compute the pairwise feature interactions by two independent subnetworks. Branson et al. [9] introduced the combination of high and low level features to learn more discriminative representations. Wang et al. [10] showed that mid-level representation learning can be enhanced with the CNNs. Therefore, they proposed a discriminative mid-level patches with a CNNs to improve classification accuracy. In recent years, visual attention models have been widely applied to computer vision, and various studies have successfully applied it to image classification. Zheng et al. [11] proposed a multi-attention convolutional neural network that learns discriminative part localization and fine-grained feature representation for image classification. Besides, Liu et al. [12] introduced part-based attributes to guide learning more discriminative features for image classification. However, these methods primarily focus on classification of categories of one particular level, and overlook hierarchical structure information of categories.

It is an interesting research to embed the hierarchy of object categories in image recognition. Because an inherent hierarchical data structure can often be seen in image datasets, where the images are at a subordinate level, and the human visual system is already capable of basic-level visual classification before fine visual classification. Therefore, it is in line with the human way of thinking for a computer to learn the hierarchy of embedded object categories for image recognition. Srivastava et al. [13] proposed a method that augments standard neural networks with tree-based priors over the classification parameters. They exploited class hierarchy to transfer knowledge among similar classes for transfer learning. Although [13] achieves a good performance, it does not use the hierarchical nature within the layers. Yan et al. [14] introduced hierarchical deep CNNs (HD-CNN) by embedding deep CNNs into a two-level category hierarchy. Specifically, the HD-CNN uses a coarse category classifier to separate easy categories, and a fine category classifier to separate difficult categories. One limitation of the HD-CNN is that it requires the coarse and fine category components to be pre-trained, which is quite time-consuming. Zhu et al. [8] proposed a branch convolutional neural network for image recognition. Similar to our approach, they use hierarchical label tree to train a multiple branch networks. Although this approach achieves good results, it does not combine the specific information at different levels, which is not sufficient to represent the object. Liu et al. [15] build a Confusion Visual Tree (CVT) based on confused semantic level information to identify the confused categories, which can pay more attention on confused categories, thereby effectively boosting the classification accuracy. Chen et al. [16], built a new Hierarchical Semantic Embedding (HSE) network. It predicts the score vector for each level, and leverages the predicted score vector of the coarse level to guide learning fine level feature representation. Some works [17–24] also adopt class hierarchy to combine different models for image classification. Although these existing approaches have achieved a substantial improvement on image classification, one potential problem is that they difficult to capture more specific information (e.g. context information) at different levels of granularity, and this problem may seriously limit their classification accuracy.

Based on the above works, our HB-CNN integrates the category hierarchy into the CNNs to progressively regularise label prediction and guide representation learning. In addition, it exploits a connectivity pattern to smoothly pass hierarchical conceptual information, and embeds bilinear module to model subtle visual differences.
3 | ARCHITECTURE OF THE PROPOSED HB-CNN

In this section, we first review Bilinear Convolutional Neural Network (B-CNN). Then, we introduce the Hierarchical Bilinear Convolutional Neural Network (HB-CNN) architecture. Finally, we discuss possible training strategies of HB-CNN.

3.1 | Bilinear convolutional neural network

The bilinear convolutional neural network (B-CNN) is proposed by Lin et al. [7]. B-CNN represents an image as a pooled outer product of features derived from two feature extractors. It can model local pairwise feature interactions, and generalise various orderless texture feature descriptors. The overall architecture of B-CNN is illustrated in Figure 1. The B-CNN model is expressed by the following equation:

\[ a = (F_1, F_2, P, C) \]  

(1)

Here, \( F_1 \) and \( F_2 \) represent the feature functions, \( P \) is a pooling function, and \( C \) denotes a classification function. The feature outputs are combined at each location using the matrix outer product. The bilinear combination of \( F_1 \) and \( F_2 \) at a location \( l \) can be expressed by using the following expression:

\[ \text{Bilinear}(l, l, F_1, F_2) = F_1(l, I)^T F_2(l, I) \]  

(2)

Then, the sum pooling aggregates the bilinear combination of features across all locations (\( L \) represents the set of locations) in the image to obtain a global image representation \( \Phi(I) \).

\[ \Phi(I) = \sum_{l \in L} F_1(l, I)^T F_2(l, I) \]  

(3)

The resulting bilinear vector \( x = \Phi(i) \) is then passed through signed square root step \( \gamma = \text{sign}(\gamma) \sqrt{|\gamma|} \), followed by \( l_2 \) normalisation \( \phi = \gamma / \|\gamma\|_2 \). More details of the B-CNN can be found in Ref. [7].

3.2 | Hierarchical Bilinear Convolutional Neural Network

The overall architecture of Hierarchical Bilinear Convolutional Neural Network (HB-CNN) is illustrated in Figure 2 and a corresponding label tree is in Figure 3. The label tree can usually be generated by unsupervised methods [25] or manually constructed based on visual similarity.

The HB-CNN constructs a network with internal output branches on the existing CNN (e.g. VGG16 [2]). Each level in the label tree of the data is captured by at least one branch, and each branch outputs a prediction at the corresponding level in the label tree (Figure 3, shown in same colour). On the top of each branch, fully connected layers and a softmax layer are
used to produce the output. To smoothly pass hierarchical conceptual information and embed multi-granularity semantic features, a connectivity pattern is proposed to connect the features of multiple output branches. On the top of last branch, a bilinear module [7] is embedded, which can model subtle visual differences to support visual-similar objects recognition.

### 3.2.1 Connectivity pattern

To smoothly pass hierarchical conceptual information and encourage feature reuse, we propose a connectivity pattern (CP): for each output branch layer, the feature-maps of preceding branch layer and its own feature-maps are used as input into subsequence branch layer. This connectivity pattern is illustrated in Figure 2. Consequentially, the $k^{th}$ level branch receives the feature-maps expressed as follows:

$$x_k = \text{Concat}(x_{k-1}, FC(b_k)) \quad (4)$$

where $x_{k-1}$ refers to the concatenation of the feature-maps produced in branch layers $k - 1$, $b_k$ represents the network output of the $k^{th}$ block (see Figure 2), and $FC$ is the fully connected layer. We define $\text{Concat}()$ as a concatenate operation. In recent years, some works [26,27] have also used ideas similar to connectivity pattern, but they are just to pass the low-level features to the high-level. Different from these methods, the proposed connectivity pattern passes coarse-grained features to fine-grained features, thereby enhancing the distinguishability of features. For instance, for a three-branch HB-CNN, an image of a *ship* will contain a hierarchical label of \{*transport*, *water*, *ship*\}. When the image is fed into HB-CNN, the network first learns the information at the Transport, Water and Ship levels. Then, the connectivity pattern is used to integrate these different granular information to boost the classification accuracy.

### 3.2.2 Network details

We implement our framework based on the VGG16 [2]. The HB-CNN model uses VGG16 as base network to construct a network with internal output branches. The exact network configurations are shown in Table 1. Specifically, we implement the block 1 with the preceding two convolutional layers of the VGG16 and a pooling layer, and on the top of first branch, we build an additional fully connected layer and a softmax layer to produce a coarse prediction. The following structure is similar to block 1. It is worth noting that after block 5, we embed the bilinear module (see Table 1). Note that the HB-CNN with the bilinear module (B-CNN [7]) removed is called Hierarchical Convolutional Neural Network (H-CNN).

### 3.2.3 Loss Function

Cross-entropy loss is used as the loss of each branch of HB-CNN. The cross-entropy loss of the $i^{th}$ sample on the $k^{th}$ level in the label tree is expressed by the following equation:

$$L_{(k,i)} = -\log \left( \frac{e^{y_i}}{\sum_k e^{y_i}} \right) \quad (5)$$
where \( f_i \) is the \( i^{th} \) element in the class scores vector \( f \) from the last layer of the model. The final loss is defined in equation (6):

\[
L_j = \sum_{k=1}^{K} -W_k \log \left( \frac{e^{f_k}}{\sum_j e^{f_j}} \right)
\]  

(6)

\( K \) represents the number of levels in the label tree, and \( W_k \) is the loss weight of the \( k^{th} \) level. The loss function takes into account the loss of each branch layer, ensuring that the structural priors play an internal guiding role for the entire model, making it easier to flow the gradients back to the shallow layers.

### 3.3 Training strategies

Value \( W_k \) in Equation (6) defines how much contribution a level makes to the final loss function. We use a hierarchical training strategy to jointly optimise multiple losses, and each branch uses different supervisory information to train the network. The supervision labels of each branch shall be determined according to the label tree structure of the dataset. There are three different patterns of this hierarchical training strategies. (1) Branch-by-branch training strategy (BBT-strategy): we use the parameters trained by the previous branch as the initialisation parameters, and then use the supervision labels to fine-tune the module of the latter branch. For example, for a three-branch HB-CNN structure, the loss weights are first set to [1, 0, 0] and changed to [0, 1, 0] after 50 epochs and assigned as [0, 0, 1] after 70. This training strategy makes each branch to be fully trained, but ignores the connection of information between tasks. (2) Multi-branch joint training strategy (MBJ-strategy): According to importance of the task, a fixed weight is assigned to each loss, and all branches are trained simultaneously. For example, for a three-branch HB-CNN structure, the loss weights are first set to [0.1, 0.1, 0.8]. This training strategy can be regarded as a multi-task learning strategy, which can jointly train the related classifiers among them to enhance objects discrimination power. (3) Multi-branch dynamic joint training strategy (MBD-strategy): this training strategy is similar to the multi-branch joint training strategy except that it dynamically modifies the loss weight distribution while training HB-CNN. For example, for a three-branch HB-CNN structure, the loss weights are first set to [0.8, 0.1, 0.1] and changed to [0.1, 0.8, 0.1] after 50 epochs and assigned as [0.1, 0.1, 0.8] after 70. Throughout the training process, it shifts loss weight distribution from coarse-level to fine-level and guides the HB-CNN model to pay more effort to learn particular level at different training stages, thus paying attention to more subtle regions.

### 4 EXPERIMENTAL RESULTS AND ANALYSIS

This section describes our experimental results and analysis for algorithm evaluation over multiple image sets.

**Implementation Details:** We utilise VGG16 as base network, and initialise the parameters with pre-trained parameters on ImageNet, and then fine tune it on CIFAR-10, CIFAR-100, and ‘Orchid’ Plant Image Set. The HB-CNN is constructed based on the classical VGG16 (second column in Table 1) and its corresponding branch networks (third and fourth columns in Table 1). For network training, we using stochastic gradient decent (SGD) optimiser with momentum 0.9. In all experiments, we will step-wise decompose our approach to reveal the effect of each component.

#### 4.1 Dataset

##### 4.1.1 CIFAR-10

The CIFAR-10 dataset [23] contains 10 classes of \( 32 \times 32 \) RGB images. The 60,000 images are divided into training and test sets, which have 50,000 and 10,000 images, respectively. For the CIFAR-10 dataset, we manually construct a label tree (see Figure 3).

##### 4.1.2 CIFAR-100

The CIFAR-100 dataset [28] has 100 classes containing 600 images each. The CIFAR-100 is divided into 20 superclasses and we manually grouped into eight coarser classes to provide richer information.

##### 4.1.3 Orchid Plant Image Set

We have crawled images which included 51 plant species (object classes) in the ‘Orchid’ family, where 32,064 plant images are used to train and other 7894 images are used for testing. Figure 4 shows some examples from ‘Orchid’ Plant Image Set. The coarse-grained level corresponds to the genus of ‘Orchid’, which contains eight classes. The fine-grained level is the target level corresponding to the ‘Orchid’ species, containing 51 target classes. The category information contained in the ‘Orchid’ Plant image set is shown in Table 2.

#### 4.2 HB-CNN on CIFAR-10 and CIFAR-100

The HB-CNN model is constructed with VGG16 as baseline model and adds five additional branches to output predictions (see Table 1). In Table 1, the block \( FC - 128 \), \( FC - 128 \),
The coarse 1 branch, the block \{FC - 256, FC - 256, FC - c1\} is the coarse 2 branch, the block \{FC - 1024, FC - 1024, FC - c2\} is the coarse 2 branch, the block \{FC - 1024, FC - 1024, FC - c3\} is the fine branch, and the block \{B - CNN, FC - c3\} is the fine branch.

For HB-CNN, both model’s initial learning rates are set to be 0.003 and decrease to 0.0005 after 40 epoch and 0.0001 after 50 epoch. The loss weight distribution for the HB-CNN is denoted as \{A1, A2, A3, A4, A5\} where A1, A2, A3, A4, A5 refer to the loss weight of the coarse 1 branch, coarse 2 branch, coarse 2 branch, fine branch and fine branch, respectively.

4.2.1 Comparison of Training Strategies

In the experiment, we use three different strategies to verify the advantages of hierarchical training. Table 3 shows the loss weight distribution. It can be seen from Table 3 that the hierarchical training strategy (MBD)-strategy can effectively boost the classification accuracy. One possible reason is that it can be regarded as a multi-task learning strategy, which can jointly train the related classifiers among them to enhance objects discrimination power. Further, it guides the H-CNN model to pay more effort to learn particular level at different training stages, thus paying attention to more subtle regions.

4.2.2 Justifications of Number of Branches

In order to study the effect of branch number on classification accuracy, we compared three-branch H-CNN structure with five-branch H-CNN structure. The three-branch H-CNN is constructed based on VGG16 (second column in Table 1) and its corresponding branch networks (fourth column in Table 1). Figures 5 and 6 show results on the CIFAR-10 and CIFAR-100, respectively. Compared with the three-branch H-CNN, the five-branch H-CNN can significantly improve the classification accuracy. We attribute this to two factors: (1) From the

**Figure 4** Some samples from ‘Orchid’ Plant image set

**Table 2** Label hierarchy shown as the family, genus and species. Our dataset contains orchid of 51 different species under eight families

| Family | Genus         | Species                                                                 |
|--------|---------------|-------------------------------------------------------------------------|
| Orchid | Angulaoa      | Angulaoa brevilabris, Angulaoa clowesii, Angulaoa eburnea, Angulaoa hohenlohei, Angulaoa ruckeri, Angulaoa uniflora, Angulaoa virginialis |
| Orchid | Ascocentrum   | Ascocentrum ampullaceum, Ascocentrum aurantiacum, Ascocentrum christosonianum, Ascocentrum garayi, Ascocentrum miniatum |
| Orchid | Brasiliorchis | Brasiliorchis chrysanthha, Brasiliorchis gracilis, Brasiliorchis marginata, Brasiliorchis phoenicanthea, Brasiliorchis picia, Brasiliorchis porphyrostele, Brasiliorchis schunkeana, Brasiliorchis ubutubuna |
| Orchid | Brassavola    | Brassavola acaulis, Brassavola ceboletta, Brassavola cucullata, Brassavola flagellaris, Brassavola nodosa, Brassavola perrini, Brassavola subulifolia, Brassavola tuberculata |
| Orchid | Brassia       | Brassia arachnoidea, Brassia arvaguera, Brassia aurantiaca, Brassia caudata, Brassia chloroleuca, Brassia keiliana, Brassia maculata, Brassia verrucosa |
| Orchid | Calochilus    | Calochilus campestris, Calochilus herbarum, Calochilus paludosus, Calochilus platychilis, Calochilus robertsonii |
| Orchid | Calopogon     | Calopogon barbatu, Calopogon multiflorus, Calopogon oklahomensis, Calopogon pallidus, Calopogon tuberosus |
| Orchid | Cephalanthera | Cephalanthera austiniata, Cephalanthera damasonium, Cephalanthera falcata, Cephalanthera longifolia, Cephalanthera rubra |
perspective of training strategy: when the network is very deep, updating lower-level parameters becomes very difficult. Compared to the three-branch H-CNN, the five-branch H-CNN updated the parameters on the lower level earlier, which alleviates this problem. (2) From the perspective of connectivity pattern: the five-branch H-CNN can introduce more coarse-grained features to fine-grained features, which may be helpful for image classification.

The performance of models trained on CIFAR-10 are shown in Table 4. The baseline model (VGG16) gets 88.11% accuracy while its corresponding H-CNN reaches 89.15%. This phenomenon confirms that using multi-granularity semantic concepts can effectively regularise semantic space and provide extra guidance to focus on more subtle regions. The B-CNN improves the performance from 88.11% to 88.40%. This shows that B-CNN can learn more discriminative representations. The connectivity pattern boosts the classification performance by a margin of 0.37% (from 88.78% to 89.15%). A possible reason is that the connect pattern embeds multi-granularity semantic features, which makes the network to capture more differentiated information at different levels. Further, it can be seen from the experimental results that HB-CNN can further improve the classification performance. It is worth noting that herein, we designed two network structures: H-CNN and HB-CNN, where HB-CNN is the combination of H-CNN and B-CNN [7]. It can be seen from the Table 4 that both the proposed HB-CNN and H-CNN can achieve better classification performance than B-CNN [7].

The training procedures on CIFAR-100 are very similar to the ones on CIFAR-10 except the learning rate. The learning rates of all models trained on CIFAR-100 are initialised as 0.001 and drop to 0.0002 at epoch 60 and 0.00005 after epoch 70. It can be seen from the experimental results that our proposed method shows similar properties on CIFAR-10 and CIFAR-100. Figure 7 shows some examples where the H-CNN model predicts correctly but VGG16 predicts incorrectly. These results further

| Table 3: Loss weights distribution on CIFAR-10 and CIFAR-100 |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
|                                | Epochs (CIFAR-10) | Epochs (CIFAR-100) | BBT-strategy | MBJ-strategy | MBDJ-strategy |
|                                |                  |                  |               |               |               |
| -                               | 0 – 8            | 0 – 13            | [1, 0, 0, 0]   | [0.1, 0.1, 0.2, 0.8, 0.8] | [0.8, 0.05, 0.05, 0.05, 0.05] |
| -                               | 8 – 18           | 13 – 23           | [0, 1, 0, 0]   | [0.1, 0.1, 0.2, 0.8, 0.8] | [0.05, 0.8, 0.05, 0.05, 0.05] |
| -                               | 18 – 28          | 23 – 33           | [0, 0, 0, 0]   | [0.1, 0.1, 0.2, 0.8, 0.8] | [0.05, 0.05, 0.8, 0.05, 0.05] |
| -                               | 28 – 38          | 33 – 43           | [0, 0, 0, 0]   | [0.1, 0.1, 0.2, 0.8, 0.8] | [0.05, 0.05, 0.8, 0.05, 0.05] |
| -                               | 38 – 60          | 43 – 150          | [0, 0, 0, 0]   | [0.1, 0.1, 0.2, 0.8, 0.8] | [0, 0, 0, 0, 1] |
| Accuracy(CIFAR-10)             | -                | -                | 88.65%         | 88.95%         | 89.15%         |
| Accuracy(CIFAR-100)            | -                | -                | 65.00%         | 65.02%         | 65.87%         |

**Figure 5** Compare the number of network branches on CIFAR-10. w/o CP and w/CP represent without and with Connectivity Pattern (CP) module.
verify that our method can effectively distinguish visually similar objects. Figure 8 shows some examples of H-CNN model prediction errors on the CIFAR-100 dataset. One possible reason is that these images are not only very similar but also very fuzzy, and the human visual system is difficult to distinguish, which increases the difficulty of H-CNN classification.

4.3 | HB-CNN on ‘Orchid’ Plant Image Set

The training procedure on ‘Orchid’ Plant image set is very similar to the ones on CIFAR-10. Note that we resize the image to 224 × 224 pixels before passing it through the network. The loss weight distribution for the HB-CNN is denoted as \([A_1, A_2, A_3, A_4, A_5]\) where \(A_1, A_2, A_3, A_4, A_5\) refer to the loss weight of the coarse branch, coarse branch, coarse branch, fine branch and fine branch, respectively. It has been verified that hierarchical training strategy (MBDJ-strategy) is very effective on CIFAR-10 and CIFAR-100 datasets, so we use the hierarchical training strategy to train the model. The loss weights modification scheme for HB-CNN are: epoch 1 – 20: \([0.8, 0.05, 0.05, 0.05, 0.05]\), epoch 20 – 30: \([0.05, 0.05, 0.05, 0.05, 0.05]\), epoch 30 – 40: \([0.05, 0.05, 0.05, 0.05, 0.05]\), epoch 40 – 120: \([0.05, 0.05, 0.05, 0.05, 0.05]\), epoch 120 – 200: \([0, 0, 0, 0, 1]\). The performance of models trained on ‘Orchid’ Plant image set are shown in Table 4.

### Table 4

| Approach                  | CIFAR-10 | CIFAR-100 | ‘Orchid’ |
|--------------------------|----------|-----------|----------|
| Vgg16                    | 88.11    | 62.97     | 84.02    |
| H-CNN(w/o CP)            | 88.78    | 64.65     | 84.78    |
| H-CNN(w/CP)              | 89.15    | 65.87     | 86.32    |
| B-CNN                    | 88.40    | 63.43     | 89.40    |
| HB-CNN                   | 91.75    | 66.03     | 91.10    |

4.4 | Comparison with state-of-the-art methods

In this subsection, we compare our method with eleven representative image classification methods: VGG16 [2], ResNet-101 [29], HD-CNN [14], Branch-CNN [8], Coarse-to-fine CNN [20], Tree-CNN [30], VI-CNN B [15], B-CNN [7], Highway Net [31], Capsule Net [26] and Autonomous [32]. Here, we evaluate on recognising the categories of the finest level (10 species on CIFAR-10, 100 species on CIFAR-100 and 51 subcategories on ‘Orchid’ Plant image set) as existing methods primarily report their results of this level. The classification results are quantified and compared, as illustrated in Table 5. With the exception of the results on the ‘Orchid’ Plant image set, the results of all competing methods were originally published. Note that in VGG16 and B-CNN, the authors did not calculate results on the CIFAR-10, CIFAR-100 and ‘Orchid’ Plant image set, so these are calculated by us.

As shown in Table 5, our method obtained classification accuracy of 91.75%, 66.03% and 91.10% on the CIFAR-10, CIFAR-100 and ‘Orchid’ Plant image set respectively, which outperforms all the previous approaches. This proves that the method of seamlessly integrating CNNs with multi-task learning over the hierarchical visual concept structures is superior to other methods. It is worth noting that our proposed method is inspired by [8] and achieved better
recognition accuracy than [8]. The reason may come from the following three aspects: 1) We use more branches, which can introduce more coarse-grained features into fine-grained features to help image classification; 2) The proposed connectivity pattern can smoothly pass hierarchical conceptual information and encourage feature reuse; 3) The embedded bilinear module can capture more distinguishing features to reduce the ambiguity of predictions.

To verify the generality and effectiveness of our proposed method, we use ResNet-101 as the base network. The HB-CNN (ResNet-101) is constructed based on the classification ResNet-101 and its corresponding branch networks. As shown in Table 5, our proposed method can improve the recognition accuracy from 90.61% to 92.85%, 67.61% to 69.23% and 91.68% to 94.29% on CIFAR-10, CIFAR-100 and ‘Orchid’, respectively.

- **HD-CNN**: uses a coarse category classifier to separate easy categories, and a fine category classifier to separate difficult categories. One limitation of the HD-CNN is that it requires the coarse and fine category components to be pre-trained, which is quite time-consuming.
- **Branch-CNN**: it uses hierarchical label tree to train a multiple predictions branch network. Although this approach achieves good results, it does not combine the specific information at different levels, which is not sufficient to represent the object.

| Approach       | CIFAR-10 | CIFAR-100 | ‘Orchid’ |
|---------------|----------|-----------|---------|
| Vgg16         | 88.11    | 62.97     | 84.02   |
| ResNet-101 [25] | 90.61    | 67.61     | 91.68   |
| HD-CNN [14]   | -        | 64.73     | -       |
| Branch-CNN [8] | 88.22    | 64.42     | 84.26   |
| Coarse-to-fine CNN [20] | 84.88    | 64.21     | -       |
| Tree-CNN [24] | 86.24    | 60.46     | -       |
| VT-CNN B [15] | 89.51    | 72.04     | -       |
| B-CNN [7]     | 88.40    | 63.43     | 89.40   |
| Highway Net [26] | 89.18    | 67.60     | -       |
| Capsule Net [27] | 89.40    | -         | -       |
| Autonomous [28] | 89.23    | 66.70     | -       |
| HB-CNN (VGG16) | 91.75    | 66.03     | 91.10   |
| HB-CNN (ResNet-101) | 92.85    | 69.23     | 94.29   |

- Coarse-to-fine CNN: Fu et al. [20] propose the coarse-to-fine layer by applying Bayesian techniques into the network, making the coarse-to-fine network can learning the hierarchical category tree.

**FIGURE 7** Some examples of the correct classification of H-CNN

**FIGURE 8** Some examples of the error classification of H-CNN

**TABLE 5** Comparison with existing image classification methods on CIFAR-10, CIFAR-100 and ‘Orchid’ Plant image sets. ‘-’ indicates results that are not presented here.
• Tree-CNN: Roy et al. [24] propose an adaptive hierarchical network structure composed of DCNNs that can grow and learn as new data becomes available.
• VI-CNN B: Liu et al. [15] build a Confusion Visual Tree (CVT) based on confused semantic-level information to identify the confused categories, which can pay more attention on confused categories.
• B-CNN: B-CNN [7] represents an image as a pooled outer product of features derived from two feature extractors.
• Highway Net: Srivastava et al. [26] propose a architecture that uses a learning gating mechanism for regulating.
• Capsule Net: In capsule networks, the capsule is a vector whose length represents the probability that entity exist, and orientation denotes properties of the entry [27].
• Autonomous: Ma et al. [28] built a genetic DCNN designer, which can generate a DCNN architecture automatically based on the data available for a specific image classification problem.

4.5 | Analysis

Figure 9 shows the influence of different models on the improvement of classification accuracy in three datasets. The following conclusions can be drawn from the experimental results. First, the performance of the H-CNN model is always better than the corresponding baseline model, which strongly proves that hierarchical nature of CNN can be connected to the structure prior of object classes to enhance the classification effect. Second, the hierarchical training strategy significantly improves the classification accuracy, which confirms that it is very useful to activate the shallow layers of a CNN model by first learning the low-level features with the coarse-level labels. Interestingly, one can easily observe that B-CNN actually has a significant improvement on ‘Orchid’ Plant image set (see Figure 9). While on the other datasets (e.g. CIFAR-10, CIFAR-100) the improvements were negligible. One possible reason is that the image size of these datasets is very small, so bilinear module cannot sufficiently generate orderless texture features. Finally, the seamless integration of H-CNN and bilinear module can not only relate the hierarchy of CNN to the structural priors of the object classes, but also extract more discriminating information of visual-similar objects, thus significantly improving the classification accuracy.

5 | CONCLUSIONS

Herein, a multi-task learning framework called Hierarchical Bilinear Convolutional Neural Network (HB-CNN) is developed by seamlessly integrating CNNs with multi-task learning over the hierarchical visual concept structures. HB-CNN can integrate the hierarchy of CNNs with the prior of the structure of the object classes to strengthen the classification ability, and it can also leverage the bilinear module to extract more discriminative information. Furthermore, we introduce a hierarchical training strategy which enables HB-CNN model to utilise label tree as internal guides, and boosts performance significantly. Our experiments on three image sets have demonstrated that our proposed approach can outperform the corresponding baseline CNN.

ACKNOWLEDGEMENTS

This work was supported by the "135 HJG" project 2018ZX01012101-007 "Radiation-immune IC microsystem".
REFERENCES

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, pp. 1097–1105 (2012)
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
3. Szegedy, C., et al: Going deeper with convolutions. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston, MA, pp. 1–9 (2015)
4. He, K., Zhang, X., Ren, S., Sun, J: Deep residual learning for image recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 770–778 (2016)
5. Deng, X., et al.: Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In: Proc. IEEE Int. Conf. Computer Vision, Seoul, Korea, pp. 1911–1920 (2019)
6. Johnson, K.E., Eilers, A.T.: Effects of knowledge and development on subordinate level categorization. Cognitr. Dev. 13(4), 515–545 (1998)
7. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, pp. 1449–1457 (2015)
8. Zhu, X., Bain, M.: B-CNN: Branch convolutional neural network for hierarchical classification (2017). arXiv preprint arXiv:1709.09890
9. Branson, S., VanHorn, G., Belongie, s., Perona, P.: Bird species categorization using pose normalized deep convolutional nets (2014). arXiv preprint arXiv:1406.2952
10. Wang, Y., Morariu, V.I., Davis, L.S.: Learning a discriminative filter bank within a CNN for fine-grained recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 4148–4157 (2018)
11. Zheng, H., et al.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, pp. 5209–5217 (2017)
12. Liu, X., et al.: Localising by describing: attribute-guided attention localization for fine-grained recognition. In: Proc. AAAI, San Francisco, CA, USA (2017)
13. Srivastava, N., Salakhutdinov, R.R.: Discriminative transfer learning with tree-based priors. In: Proc. Advances in Neural Information Processing Systems, lake Tahoe, NV, USA, pp. 2094–2102 (2013)
14. Yan, Z., et al.: HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In: Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, pp. 2740–2748 (2015)
15. Liu, Y., et al.: Visual tree convolutional neural network in image classification. In: 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, pp. 758–764 (2018)
16. Huang, G., et al.: Multi-scale dense networks for resource efficient image classification (2017). arXiv preprint arXiv:1703.09844
17. Zweig, A., Weinshall, D: Exploiting object hierarchy: Combining models from different category levels. In: 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, pp. 1–8 (2007)
18. Jia, Y., et al.: Visual concept learning: Combining machine vision and bayesian generalisation on concept hierarchies. In: Proc. Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, pp. 1842–1850 (2013)
19. Song, G.h., et al.: Two-level hierarchical feature learning for image classification. Front. Inf. Technol. Electr. Eng. 17(9), 897–906 (2016)
20. Deng, J., et al.: Large-scale object classification using label relation graphs. In: European Conf. on Computer Vision, Zurich, Switzerland, Zurich, Switzerland, pp. 48–64 (2014)
21. Fu, R., et al.: CNN with coarse-to-fine layer for hierarchical classification. IET Comput. Vis. 12(6), 892–899 (2018)
22. Seo, Y., Shin, K.: Hierarchical convolutional neural networks for fashion image classification. Expert Syst. Appl. 116, 328–339 (2019)
23. Dhall, A., et al.: Hierarchical image classification using Entailment Cone embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 836–837 (2020)
24. Chen, T., et al.: Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In: Proceedings of the 26th ACM international conference on Multimedia, pp. 2023–2031 (2018)
25. Qu, Y., et al.: Joint hierarchical category structure learning and large-scale image classification. IEEE Trans. Image Process. 26(9), 4331–4346 (2016)
26. Sabour, S., Frosst, N., Hinton, G.: Dynamic routing between capsules. Adv. Neural Inf. Process. Syst., 3856–3866 (2017)
27. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
28. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech Report (2009)
29. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
30. Roy, D., Panda, P., Roy, K.: Tree-CNN: a hierarchical deep convolutional neural network for incremental learning. Neural Netw. 121, 148–160 (2020)
31. Srivastava, R., Greff, K., Schmidhuber: Highway networks (2015). arXiv preprint arXiv:1505.00387
32. Ma, B., et al: Autonomous deep learning: a genetic DCNN designer for image classification. Neurocomputing. 379, 152–161 (2020)