Jiang *et al. EURASIP Journal on Wireless Communications and Networking* (2018) 2018:216
https://doi.org/10.1186/s13638-018-1197-z

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Tree-CNN: from generalization to specialization

Shenwang Jiang[1,2], Tingfa Xu[1,2*], Jie Guo[1,2] and Jizhou Zhang[1,2]

Abstract

Traditional convolutional neural networks (CNNs) classify all categories by a single network, which passes all kinds of samples through totally the same network flow. In fact, it is quite challengeable to distinguish schooner with ketch and chair by a single network. To address it, we propose a new image classification architecture composed of a cluster algorithm and the Tree-CNN. The cluster algorithm devotes to classifying similar fine categories into a coarse category. The Tree-CNN is comprised of a Trunk-CNN for coarse classification of all categories and Branch-CNNs to treat different groups of similar categories differently. Branch-CNNs are fine-tuning based on the Trunk-CNN, which extracts the special feature map of image and divides it into fine categories. But Branch-CNNs bring extra computation and are hard to train. To address it, we introduce adaptive algorithm to balance the heavy computation and accuracy. We have tested Tree-CNNs based on CaffeNet, VGG16, and GoogLeNet in Caltech101 and Caltech256 for image classification. Experiment results show the superiority of the proposed Tree-CNN.

Keywords: Convolutional neural network, Tree-CNN, Coarse category, Generalization, Specialization

1 Introduction

Humans glance at an image and then recognize objects in the image, which is a generalization to specialization progress. When we notice a cat, the process that we distinguish the cat from other similar categories, such as a puppy or a small leopard, is different from distinguishing it from some distinct categories like an airplane or a bicycle. In fact, how to define a category is a troublesome problem. Take the airplane for an example, it has many sub-categories such as the helicopter, the attack plane, and the airliner. We can serve the airplane as a whole category or divide it into two categories, namely the military aircraft and the airliner, and there are still more ways to divide them.

Heretofore, most image classification systems process cats, leopards, and airplanes in a same model with totally the same parameters. The long time classification systems were concentrated on designing low-level image features, such as HOG [1] and SIFT [2], to improve the performance. During this period, the feature map of image used for classification was changeless when we designed the feature descriptor. Since designing a feature descriptor is hard and time-consuming, designing special descriptors for all similar categories in a large dataset is impossible. Recently, many superior algorithms like GoogLeNet and VGGNet have been proposed, owing to the fact that AlexNet [3] has achieved a substantial improvement in image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge. Although they have significantly improved the accuracy of classification, they inherit the same shortcoming of the former algorithms. They process the cat and the airplane with the same nets. But feature map of the image obtained from the convolutional neural networks (CNN) can change with the training dataset and the architecture of the CNN, which indulges us in obtaining special feature map to distinguish similar categories. And current CNN systems are concentrated on seeking deeper, such as GoogLeNet [4] and ResNet [5]. At the same time, some researchers have noticed that broadening the CNN is also a solution to improve its performance, with ResNeXt [6] as an example. The Tree-CNN can be seen as broadening CNN. The difference between common CNN and the Tree-CNN is presented in Fig. 1.

*Correspondence: xutingfa_bit@126.com; ciom_xtf1@bit.edu.cn
[1]School of Optics and Photonics, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, Beijing, China
[2]Key Laboratory of Photoelectronic Imaging Technology and System, 5 South Zhongguancun Street, Haidian District, Beijing, Beijing, China

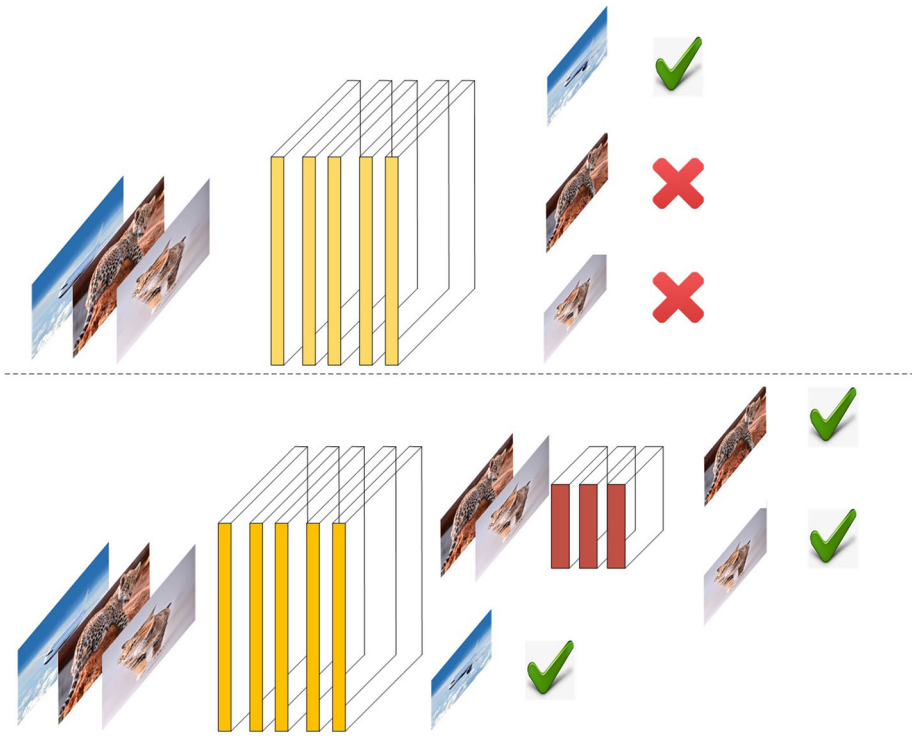

Fig. 1 The difference between common CNN (top) and the Tree-CNN (bottom). The common CNN is not sensitive to similar categories when the dataset includes many categories, but the Tree-CNN is good at distinguishing similar categories

In [7], D. Zeiler presented a view, the convolutional layers in the CNN played the analogous role of traditional feature descriptors that extracted feature map from images. And the convolutional layers are changeable compared with traditional feature descriptors. We just need to train the CNN with different image datasets. When we want to obtain a special feature map to distinguish similar categories, we only need to train the CNN with a small image dataset that constituted by the similar categories. This characteristic of the CNN makes it possible that we obtain special feature map of all similar categories.

As a topic with lasting charm, image clustering is to find a mapping of the archive images into categories such that the images in the category have a consistent visualization and summarization in content [8]. Famous image clustering such as partitioning algorithm, hierarchical algorithm, and density-based algorithm has achieved good performance. Different from traditional image cluster, which divides images into categories, our cluster is to divide similar fine categories into a coarse category. So, we need to summarize the feature map of every image in categories. And according to the summarized feature map, we divide similar categories into a coarse category.

In this paper, we propose the Tree-CNN. It is similar to a tree structural with branches and leaves. Every branch represents a set of convolutional layers that process a coarse category, and every leaf represents a fine category which is locating in the aftermost branch. The Tree-CNN brings two main problems, which are how to evaluation the similarity between categories and how to reduce the training time which increases with the number of Branchnet. To solve these problems, we propose a measure method based on the probability vector of every fine category and an adaptive algorithm to balance the heavy computation and accuracy. The adaptive algorithm links the changed layer with category similarities. Figure 2 presents the whole progress of the Tree-CNN, which is a generalization to specialization progress the same as human thinking. We implemented the Tree-CNN on Caffe [9], which is based on three CNNs (CaffeNet, VGG16, and GoogLeNet) and tested in two datasets (Caltech101, Caltech256) for image classification. Our insight considerably increases the accuracy. And there are many articles [10–14] inspired us.

To sum up, we contributed the following:

We propose a cluster algorithm that clusters fine categories into coarse categories according to similarity map. The similarity map is composed of the probability vector of every training image generated by the Trunk-CNN.

According to the progress from generalization to specialization, we propose a new framework CNN named Tree-CNN, which is constructed by the Trunk-CNN and Branch-CNNs.

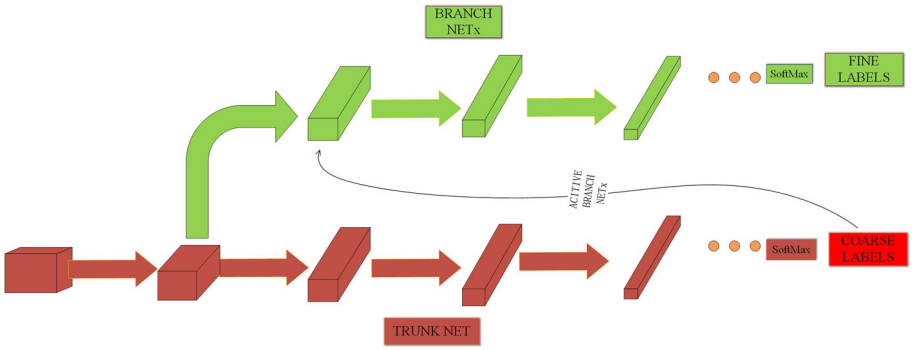

Fig. 2 Processing flow chart of the Tree-CNN. First, the Tree-CNN obtains the image's coarse category by the Trunk-Net. Second, according to the coarse category, the Tree-CNN selects a corresponding Branch-Net which obtains special feature map of the image and the fine category

To reduce the Tree-CNN training time, we present a method, in which the number of the layers in the Branch-CNN is corresponding to the similarity of fine categories in the coarse category.

1.1 Related works

1.1.1 Convolutional neural network

Convolutional neural networks (CNNs) has made great success in image processing, such as image classification [3–5, 15], object detection [16–19], remote sensing image classification [20, 21], and gender prediction [22, 23]. In 2010, Krizhevsky proposed AlexNet [3] which achieved the best performance in the ImageNet LSVRC-2010 contest and is considerably better than the second one. The performance of AlexNet stimulated the research of the CNN. Matthew D. Zeiler and Rob Fergus in 2013 introduce a novel visualization technique that gives insight into the function of convolutional layers and the operation of CNN [7]. Karen Simonyan and Andrew Zisserman show a thorough evaluation of networks [15] and propose VGG nets using an architecture with 3×3 convolution filters and increasing depth, which can significantly improve the performance. GoogLeNet [4] introduced inception which increased the width and depth of the network and brought in image scale information. ResNet [5] presents a residual learning framework that is able to deepen CNN.

1.1.2 Image clustering

Image clustering is unsupervised image classification which partitions the input images into K regions based on the contents of ImageJ. Macqueen proposed K-MEANS. According to it, the number of regions is set by humans [24]. T. Zhang introduced BIRCH algorithm [25] that built a dendrogram to cluster data.

1.1.3 Multi-branch convolutional networks

Multi-branch convolutional network has become a trend of CNN. Inception module, which is introduced by GoogLeNet [4] and constructed by multi-scale convolutional filters, has gotten great success. Feature Pyramid Networks [26] brought in multi-branch which combined multi-scale feature maps to improve the accuracy of object detection. Deep Neural Decision Forests [27] is a decision forest providing the final predictions, adding multi-branch at the end of CNN. Dividing a set of filters into separate groups by ResNeXt [6] adds a new dimension named cardinality. Increasing the cardinality can improve the classification accuracy.

2 Methods

2.1 Coarse category

Many image datasets have both the fine and the coarse categories. A fine category is constructed by accurate species such as the dog, which are similar in appearance yet different in color or texture. But a coarse category is constructed by some fine categories that may share the same characteristic. Members of a coarse category may have quite different appearance. For example, polar bear and *Carcharodon carcharias* do not look like each other, but they both belong to the carnivore. But image classification is completely based on the content of the image, the image taken by camera records, and the appearance of objects. Most of the misclassification was caused by similar appearance among fine categories, such as the cat and the small leopard. To solve the problem, we need an image dataset constructed by categories in which objects share similar appearance to train special CNN, which amplifies the difference of appearance.

Because image classification is completely based on the appearance, the probability vector $P_i = [p_{i1}, p_{i2}, \cdots, p_{in}]$ $(i = 1, 2, \cdots, n)$ outputted from the Trunk-net represents the similarity in the appearance of the ith fine category to all the other fine categories. Finally, we can obtain a square matrix $P = [P_1, P_2, \cdots, P_n]^T$ which is constituted by the similarity vector P_i. But the similarity factor p_{ij} between the ith fine category and the jth fine category is not equal

to the similarity factor p_{ji} between the jth fine category and the ith fine category. So, we chose the mean of them s_{ij} that represents the similarity factor between the ith fine category and the jth fine category.

$$s_{ij} = \frac{p_{ij} + p_{ji}}{2}, i, j = 1, 2, \cdots, n \tag{1}$$

We can obtain a symmetric matrix S that represents the similarity of each fine category. According to the symmetric matrix S, we classify fine categories to a coarse category when the similarity factor is greater than the threshold T. The flow chart of classifying fine categories into coarse categories is presented in Fig. 3. And the pseudocode is presented in Algorithm 1.

Algorithm 1 Clustering fine categories into coarse categories

Require: the symmetric matrix S; the threshold T;
Ensure:

```
Initialize label =[ l_0, l_1, ..., l_n] (l_i = 0, i = 1, 2, ..., n);
l_i represents ith fine category has been categorized into one coarse category;
while sum_i^n l_i != n do
    Initialize currentCoarse =[ k] (min_k(l_k = 0, k = 1, 2, ..., n))
    Initialize flag = 1 (if flag = 0, which represents the current coarse class has been categorized)
    while flag do
        templabelSum1 = sum_i^n l_i
        for iincurrentCoarse do
            if s_ij(j = 1, 2, ... n) > T then
                jth fine class categorized current coarse class
                l_j = 1
                currentCoarse =[ currentCoarse, j]
            end if
        end for
    end while
end while
return currentCoarse;
```

2.2 The layers need to be changed

The Tree-CNN has a Trunk-Net to classify the image to a coarse category and some Branch-Nets to classify the image to a fine category. The output of every layer in convolutional layers can be treated as a feature map, the special feature map extracted by Branch-Net roots in the generalization feature map extracted by Trunk-NET. And because of the heavy computation of CNN, Trunk-Net and Branch-Nets share shallow convolutional layers that can reduce the computation in back propagation. It is a critical problem where to add a Branch-Net to maximize the accuracy and minimize the computation. We introduce adaptive algorithm. According to that, the layers that need to be changed are related with the number and the similarity of fine categories in a coarse category, because the last layers can extract the high level information. The layers need to be changed by counting from the back to front.

$$cl = \alpha * \sum_{i}^{t} \sum_{j}^{t} s_{ij} + \beta, s.t. MinL < cl \leq MaxL \tag{2}$$

where the cl denotes the layers that need to be changed, the t denotes the number of fine categories in a coarse category, and the s_{ij} denotes the similarity between the ith fine category and the jth fine category. α and β are constant coefficient. $MinL$ and $MaxL$ represents the minimum and the maximum of layers that need to be changed, respectively.

2.3 Tree-CNN

The $P(x_t, \text{Trunk})$ represents the probability of the input image being classified into the correct tth fine category by Trunk-Net. The $P(C_i, \text{Trunk})$ represents the probability of the input image being classified into the correct ith coarse category by the Trunk-Net. And the $P(x_t, \text{Branch})$ represents the probability of the input image being classified into the tth correct fine category by Branch-Net. We assume $P(x_t, \text{Branch}) > P(x_t, \text{Trunk})$, because the Branch-Net is designed to distinguish the tth fine category from other similar categories, which is different from the Trunk-Net. And similar categories are divided into a coarse category; we assume the $P(C_i, \text{Trunk})$ is closed to 1. So, the probability of the Tree-CNN $P(x_t, \text{Branch})P(C_i, \text{Trunk})$ classifying the image into the correct fine category is greater than the probability of the original net $P(x_t, \text{Trunk})$.

Firstly, we fine-tune a Trunk-CNN using fine category image dataset based on famous framework CNN such as CaffeNet [9], VGG16, and GoogLeNet, which has been trained in ILSVRC2012. Secondly, we obtain similarity map of fine categories generated by the Trunk-CNN with all training images. And according to the similarity map, we divide fine categories into a set of coarse categories with Algorithm 1. Thirdly, we calculate which layers need to be changed by Eq. 2. Finally, we fine-tune all Branch-Net based on Trunk-Net using corresponding coarse category image dataset and which layer needs to be changed.

3 Results and discussion

Humans hand-pick the categories when they built an image dataset. The categories should represent a wide range of artificial and natural objects in various conditions and be relatively independent of each other. It aims to

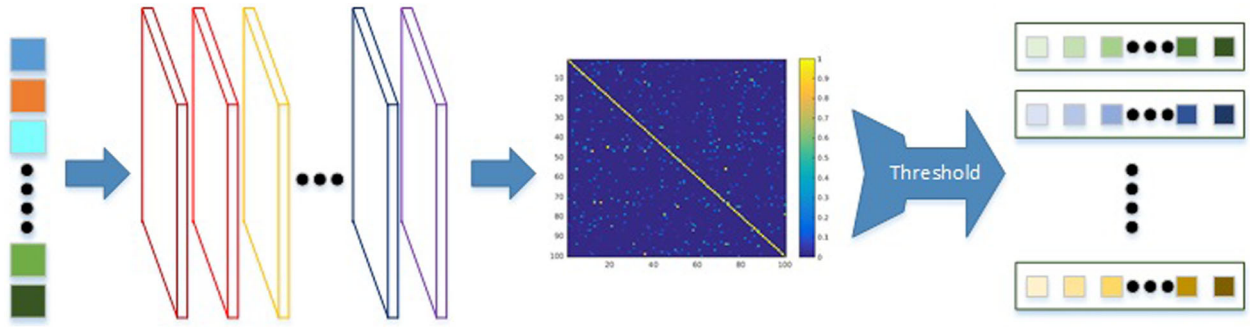

Fig. 3 The flow chart of obtaining the coarse categories. Firstly, Trunk-NET generates the probability of every training image. Secondly, Eq. 1 generates similarity map with the probability of every training image. Thirdly, according to the similarity map, Algorithm 1 engenders coarse categories with threshold T

increase the difference between categories, but some similar objects have been added in a dataset purposely, aiming to verify the capability of distinguishing similar objects. Crocodile head, schooner, and chair in Caltech101 dataset are closely related to the crocodile, the Windsor chair, and the ketch, respectively. the Airplane-101, the car-side-101, the faces-easy-101, the greyhound, the tennis-shoe, and the toad in Caltech256 dataset are closely related to the fighter-jet, the car-tire, the people, the dog, the sneaker, and the frog, respectively. For this reason, the Tree-CNN can significantly improve the accuracy of classification on caltech101 and caltech256.

3.1 Result and discussion on Caltech101

Caltech101 [28] image dataset has 9144 images and 102 categories. We removed the background category, given that the background images have no common feature in terms of their appearance. Every category has 40∼80 images with the size of roughly 300 × 200 pixels. We divided the dataset into a train dataset and a test dataset. The train dataset is composed of the top 80% images of every category sorted by filename, and the test dataset is composed by the rest of the images.

Figure 4 presents the relationship between the number of the coarse category and the threshold T in Algorithm 1 on Caltech101. In Fig. 4, we know that the number of the coarse category is positively correlated with the threshold T. The curves of VGG16 and GoogLeNet have a similar trend, which rise rapidly from threshold 0.08 and begin to flatten out from threshold 0.6. The curve of CaffeNet rises rapidly from threshold 0.2 and changes to flatten out from threshold 0.6. By our definition, the inferior value is the threshold where the curve begins to rise, and the superior value is the threshold where the curve begins to flatten.

From the accuracy of the Tree-CNN, we can know the following points. When the threshold T is greater than the superior value or less than the inferior value, the accuracy of the Tree-CNN is close to the original CNN. Because when the threshold T is greater than the superior value or less than the inferior value, the number of coarse categories is close to 1 or to the number of fine categories. Between the superior and inferior value, the Tree-CNN is equivalent to the original CNN. When the threshold T is greater than the inferior value and less than superior value, the Tree-CNN outperforms the original CNN. And between the superior and inferior value, the accuracy of the Tree-CNN shows a general trend from rise to decline. The accuracy of the Tree-CNN based on CaffeNet rises from threshold 0.05 and starts to decline from threshold 0.2. The accuracy of the Tree-CNN based on VGG16 ascends from threshold 0.05 and begins to descend from threshold 0.15. The accuracy of the Tree-CNN based on GoogLeNet goes up from threshold 0.07 and begins to descend from threshold 0.10.

Our result on Caltech101 is summarized in Table 1. Compared with the original NET, the Tree-CNN can improve the accuracy visibly. The accuracy of original CaffeNet is 92.3%, and we improve it to 92.9%. The accuracy of original VGG16 is 88.3%, and we increase it to 92.4%. The accuracy of original GoogLeNet is 95.0%, and we raise it to 96.5%. The accuracies have a problem that the VGG16 should have better accuracy than CaffeNet, which is not applicable in our result. We think it is because the quantity of the images in Caltech101 is small, which leads to over-fitting with VGG16.

To analyze the ability of the Tree-CNN, we compile statistics concerning the accuracy of the Tree-CNN and the original net in every category, which is presented in Fig. 5. And in Table 2, we list some categories whose accuracy is substantially improved by the Tree-CNN. The accuracy of all the Tree-CNN is higher than the original CNN. In Fig. 5a, we can know that the accuracy of the Tree-CNN based on CaffeNet has been significant by 0.6%. In the original CaffeNet, the metronome, the saxophone, the mayfly, and the chair are always misclassified to the pyramid, the ceiling fan, the dragonfly, and the Windsor chair, respectively. Thanks to the Tree-CNN based on CaffeNet, the accuracy of the metronome, the

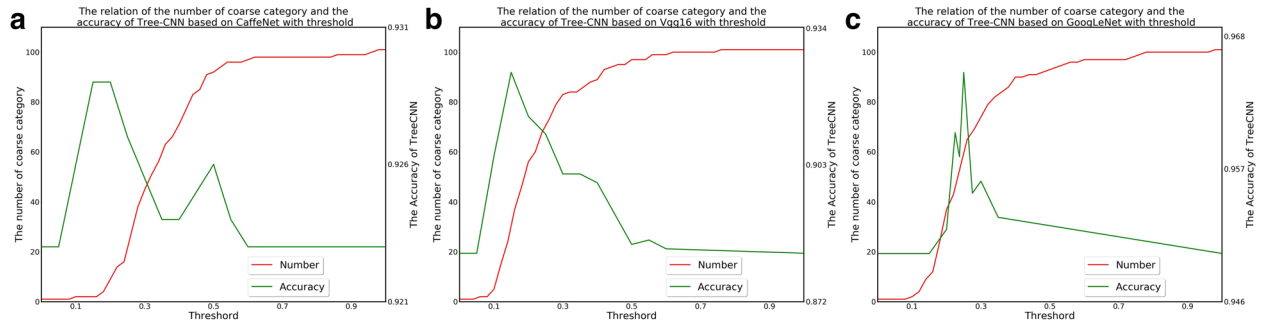

Fig. 4 The relationship between the number of the coarse category and the accuracy with the threshold. **a**–**c** CaffeNet, VGG16, and GoogLeNet. The red line represents the relationship between the number of the category and the threshold *T*. The green line represents the relationship between the accuracy of the Tree-CNN and the threshold *T*

saxophone, the mayfly, and the chair has been improved by 16.6, 14.3, 14.3, and 8.3%, respectively. In Fig. 5b, we can know that the accuracy of the Tree-CNN based on VGG16 has been remarkably improved by 4.1%. In the original VGG16, the elephant, the ketch, the kangaroo, and the octopus are always misclassified to the rhino, the schooner, the okapi, and the starfish, respectively. Owing to the Tree-CNN based on VGG16, the accuracy of the elephant, the ketch, the kangaroo, and the octopus has been raised by 91.7, 68.2, 35.3, and 33.3%, respectively. In Fig. 5c, we can know that the accuracy of the Tree-CNN based on GoogLeNet has been improved by 1.5%. In the original VGG16, the beaver, the crocodile_head, the saxophone, and the octopus are always misclassified to the cougar_body, crocodile, ceiling_fan, and starfish, respectively. Due to the Tree-CNN based on GoogLeNet, the accuracy of the beaver, the crocodile_head, the saxophone, and the octopus has been increased by 44.5, 30.0, 28.6, and 16.7%, respectively.

Figure 6 shows some images in the Caltech101 test image dataset, which are misclassified by the original CNN and corrected by the Tree-CNN. A chair in front of a fence is displayed in the first picture, which is misclassified to a Windsor chair. The second image is a crocodile going ashore, which is misclassified to a crocodile head, because part of the crocodile body is immersed in the water. A scorpion in the third image is misclassified into a crayfish, because the tail of the scorpion can not be easily identified. A ketch in the fourth image is misclassified to a schooner, because they resemble each other. A ketch has

Table 1 The accuracy of the Tree-CNN and the original network on Caltech101, which includes CaffeNet, VGG16, and GoogLeNet

CaffeNet			VGG16			GoogLeNet		
Class	Origin (%)	Tree (%)	Class	Origin (%)	Tree (%)	Class	Origin (%)	Tree (%)
All	92.30	92.90	All	88.30	92.40	All	95.00	96.50
Metronome	83.30	100.00	Elephant	0.00	91.70	Beaver	44.40	88.90
Saxophone	42.90	57.10	Ketch	18.20	86.40	Crocodile_head	70.00	100.00
Mayfly	42.90	57.10	Kangaroo	52.90	88.20	Saxophone	71.40	100.00
Cannon	75.00	87.50	Octopus	16.70	50.00	Octopus	83.30	100.00
Headphone	75.00	87.50	Strawberry	66.70	100.00	Binocular	83.30	100.00
Beaver	55.60	66.70	Scorpion	68.80	100.00	Platypus	83.30	100.00
Emu	90.00	100.00	Okapi	57.10	85.70	Brontosaurus	87.50	100.00
Elephant	91.70	100.00	Water_lily	28.60	57.10	Anchor	75.00	87.50
Soccer_ball	75.00	83.30	Lobster	50.00	75.00	Lobster	87.50	100.00
Chair	75.00	83.30	Crayfish	46.20	69.20	Pigeon	75.00	87.50
Llama	60.00	66.70	Tick	44.40	66.70	Bass	90.00	100.00
Scorpion	93.80	100.00	Crocodile	44.40	66.70	Wheelchair	90.90	100.00
Kangaroo	82.40	88.20	Beaver	22.20	44.40	Stegosaurus	90.90	100.00
Chandelier	95.20	100.00	Ceiling_fan	66.70	88.90	Cup	90.90	100.00
Watch	93.60	95.70	Crab	35.70	57.10	Elephant	91.70	100.00

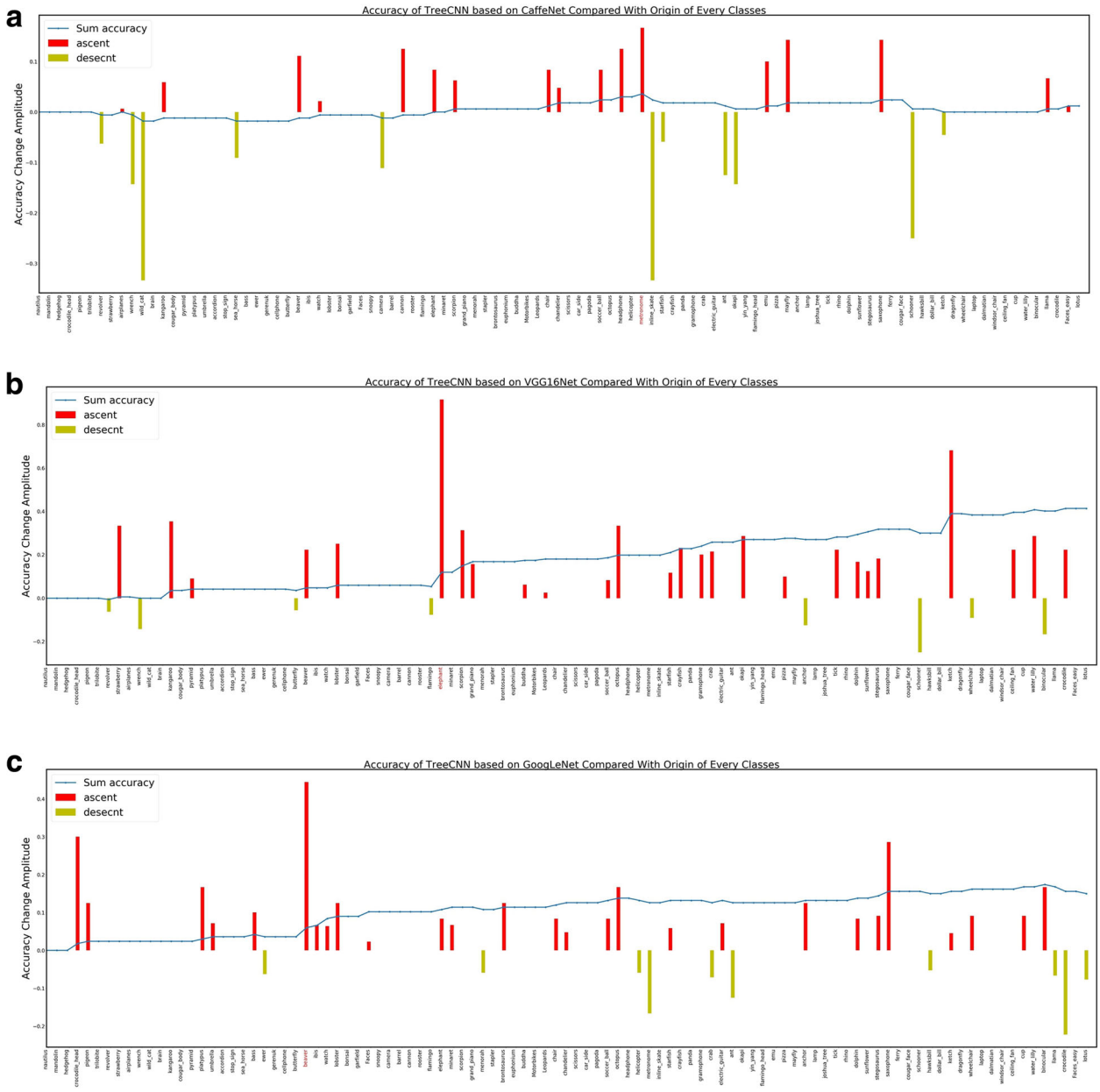

Fig. 5 Tree-CNN accuracy compared with the original CNN accuracy of every category in Caltech101. **a**–**c** CaffeNet, VGG16, and GoogLeNet. The red bar represents the accuracy of the Tree-CNN which is higher than the original CNN in this category. The yellow bar represents the accuracy of the Tree-CNN which is lower than the original CNN in this category. The curve shows that the average accuracy variation changes with the increase in categories, and the average accuracy variation had been magnified ten times

two masts with the mizzen mast stepped before the rudder head, while a schooner has two or more masts and the after masts must be the same height or greater than the foremast.

3.2 Result and discussion on Caltech256

Caltech256 [29] image dataset has 30,608 images and 257 categories, of which every category has a minimum of 80 images. As in Caltech101, we removed the background category. The train dataset and the test dataset are also not distinguished in Caltech256, so we divided the dataset into the above two parts.

Figure 7 presents the relationship between the number of the coarse category and the threshold T in Algorithm 1 on Caltech256. In Fig. 7, we know that the number of the coarse category is positively correlated with the threshold T. The curve of CaffeNet rises rapidly from threshold 0.09 and changes to flatten from threshold 0.45. The curve of

Table 2 The accuracy of the Tree-CNN and the original network on Caltech101, which includes CaffeNet, VGG16, and GoogLeNet

CaffeNet			VGG16			GoogLeNet		
Class	Origin (%)	Tree(%)	Class	Origin (%)	Tree (%)	Class	Origin (%)	Tree (%)
All	71.00	74.50	All	64.40	69.50	All	80.40	82.60
Iguana	38.10	76.20	Revolver-101	5.30	84.20	Jesus Christ	41.20	76.50
Giraffe	68.80	93.80	Duck	11.80	88.20	Snail	56.50	82.60
Lathe	50.00	75.00	Touring-bike	9.50	85.70	Traffic-light	57.90	78.90
Cormorant	52.40	76.20	Homer-simpson	0.00	63.20	Fire-hydrant	63.20	84.20
Syringe	40.90	63.60	Light-house	29.70	91.90	Screwdriver	45.00	65.00
Toaster	61.10	83.30	Chopsticks	6.20	62.50	Sword	50.00	70.00
Frog	34.80	56.50	Beer-mug	0.00	55.60	Bear	55.00	75.00
Yo-yo	36.80	57.90	Comet	0.00	54.20	Eyeglasses	62.50	81.20
Sushi	47.40	68.40	Dog	31.60	84.20	Drinking-straw	18.80	37.50
Unicorn	36.80	57.90	Goldfish	27.80	77.80	Hot-dog	43.80	62.50
Porcupine	75.00	95.00	Crab-101	12.50	62.50	Sneaker	54.50	72.70
Elephant-101	73.10	92.30	Speed-boat	5.30	52.60	Basketball-hoop	70.60	88.20
Toad	71.40	90.50	Kangaroo-101	43.80	87.50	Xylophone	50.00	66.70
Gas-pump	44.40	61.10	Gorilla	40.50	83.30	Greyhound	61.10	77.80
Tambourine	61.10	77.80	Frisbee	10.50	52.60	Picnic-table	61.10	77.80

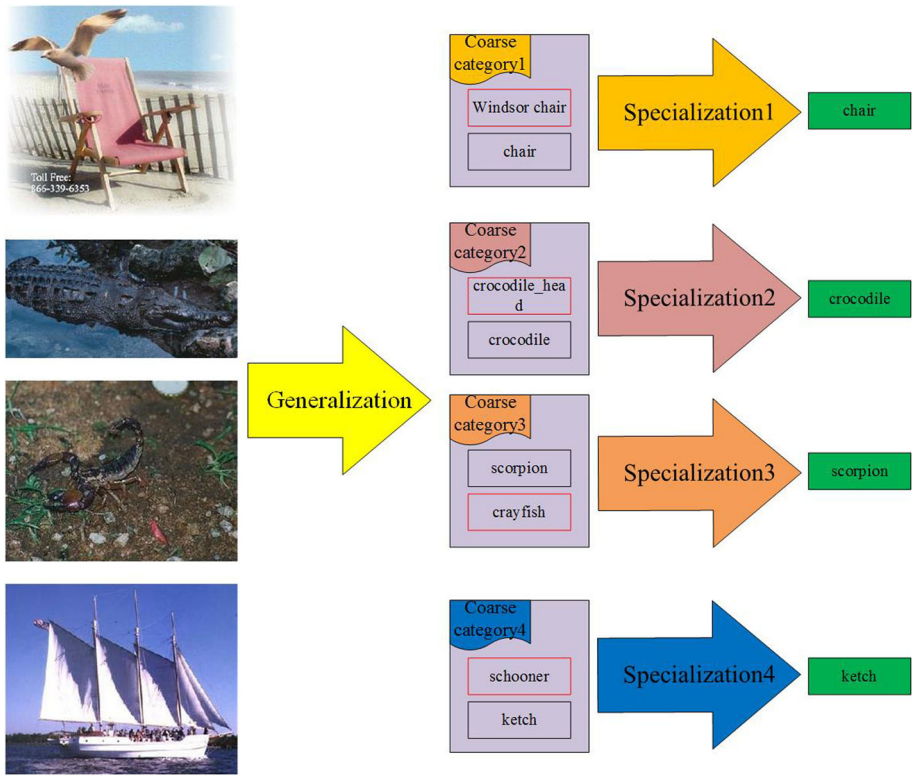

Fig. 6 Some images from Caltech101 were misclassified by the Trunk-CNN and corrected by Branch-CNNs. The red box in Coarse categories represents the output of the fine category by the Trunk-CNN. The green box represents the final output of the Tree-CNN

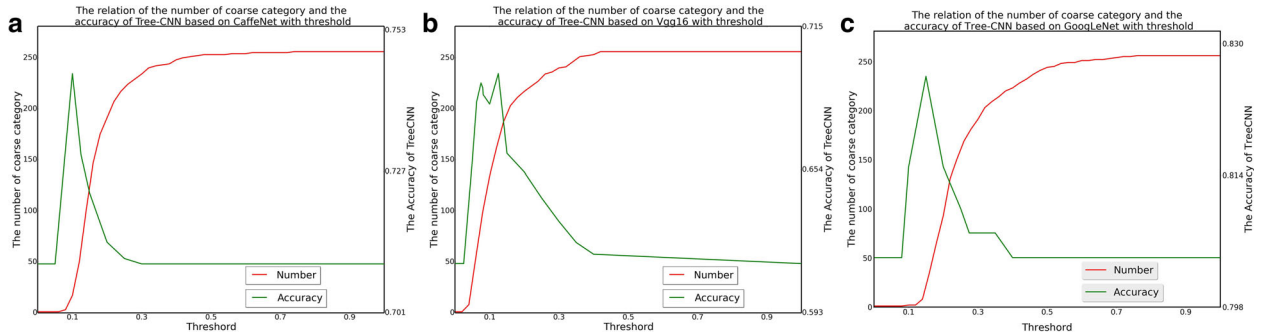

Fig. 7 The relationship between the number of the coarse category and the accuracy with the threshold *T*. **a**–**c** CaffeNet, VGG16, and GoogLeNet. The red line represents the relationship between the number of the category and the threshold. The green line represents the relationship between the accuracy of the Tree-CNN and the threshold *T*

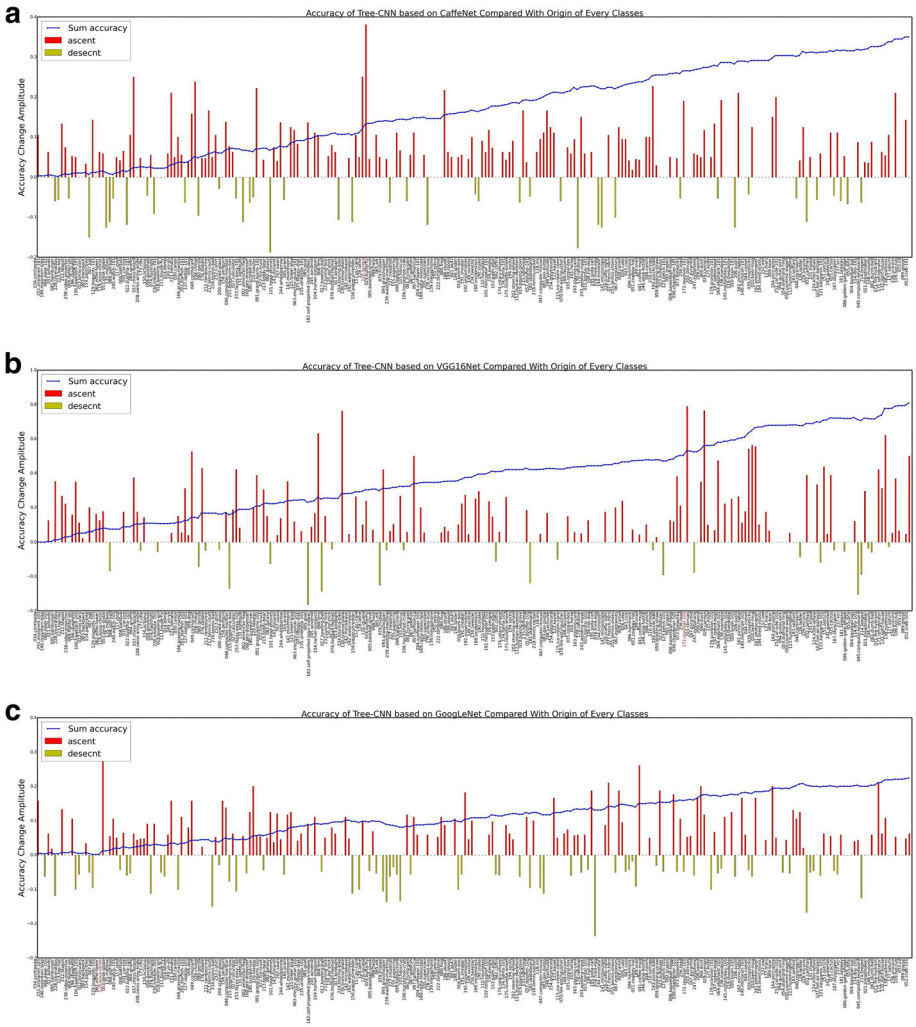

Fig. 8 The accuracy of the Tree-CNN compared with the origin of every class in Caltech256. **a**–**c** CaffeNet, VGG16, and GoogLeNet. The red bar represents the accuracy of the Tree-CNN which is higher than the original net in this category. The yellow bar represents the accuracy of the Tree-CNN which is lower than the original net in this category. The curve shows that the average accuracy variation changes with the increase in categories, and the average accuracy variation has been magnified ten times

VGG16 rises rapidly from threshold 0.04 and changes to flatten from threshold 0.4. The curve of GoogLeNet rises rapidly from threshold 0.12 and changes to flatten from threshold 0.7. Because the number of categories in Caltech256 was greater than that in Caltech101, the curve of Tree-CNN on Caltech256 rises and flattens earlier than that on Caltech101.

From the accuracy of the Tree-CNN in Fig. 7, we can know the following points. The accuracy of the Tree-CNN based on CaffeNet rises from threshold 0.05 and starts to decline from threshold 0.1. The accuracy of the Tree-CNN based on VGG16 ascends from threshold 0.025 and begins to descend from threshold 0.15. The accuracy of the Tree-CNN based on GoogLeNet rises from threshold 0.09 and falls from threshold 0.15.

Our result on Caltech256 is summarized in Table 2. Compared with the original net, the Tree-CNN can improve the accuracy visibly. The accuracy of original CaffeNet is 71.0%, and we improve it to 71.4%. The accuracy of original VGG16 is 64.4%, and we improve it to 69.0%. The accuracy of original GoogLeNet is 80.4%, and we improve it to 80.9%.

To analyze the ability of the Tree-CNN, we compile the statistics of the accuracy variation of the Tree-CNN with the original net in every class, which is presented in Fig. 8. And in Table 2, we list some categories whose accuracy is substantially improved by the Tree-CNN. The accuracy of all the Tree-CNN is better than the original net. In Fig. 8a, we can know that the accuracy of the Tree-CNN based on CaffeNet has been increased by 3.5%. In the original CaffeNet, the iguana, the giraffe, the lathe, and the cormorant are always misclassified to the toad, the goat, the floppy disk, and the penguin, respectively. Thanks to the Tree-CNN based on CaffeNet, the accuracy of the iguana, the giraffe, the lathe, and the cormorant has been improved by 38.1, 25.0, 25.0, and 23.8%, respectively. In Fig. 8b, we can know that the accuracy of the Tree-CNN based on VGG16 has been remarkably improved by 5.1%. In the original VGG16, the revolver-101, the duck, the touring-bike, and the Homer Simpson are always misclassified to the ak47, the goose, the mountain-bike, and the cartman, respectively. Owing to the Tree-CNN based on VGG16, the accuracy of the revolver-101, the duck, the touring-bike, and the Homer Simpson has been raised by 78.9, 76.5, 76.2, and 63.2%, respectively. In Fig. 8c, we can know that the accuracy of the Tree-CNN based on GoogLeNet has been remarkably enhanced by 2.2%. In the original GoogLeNet, the Jesus Christ, the snail, the traffic-light,

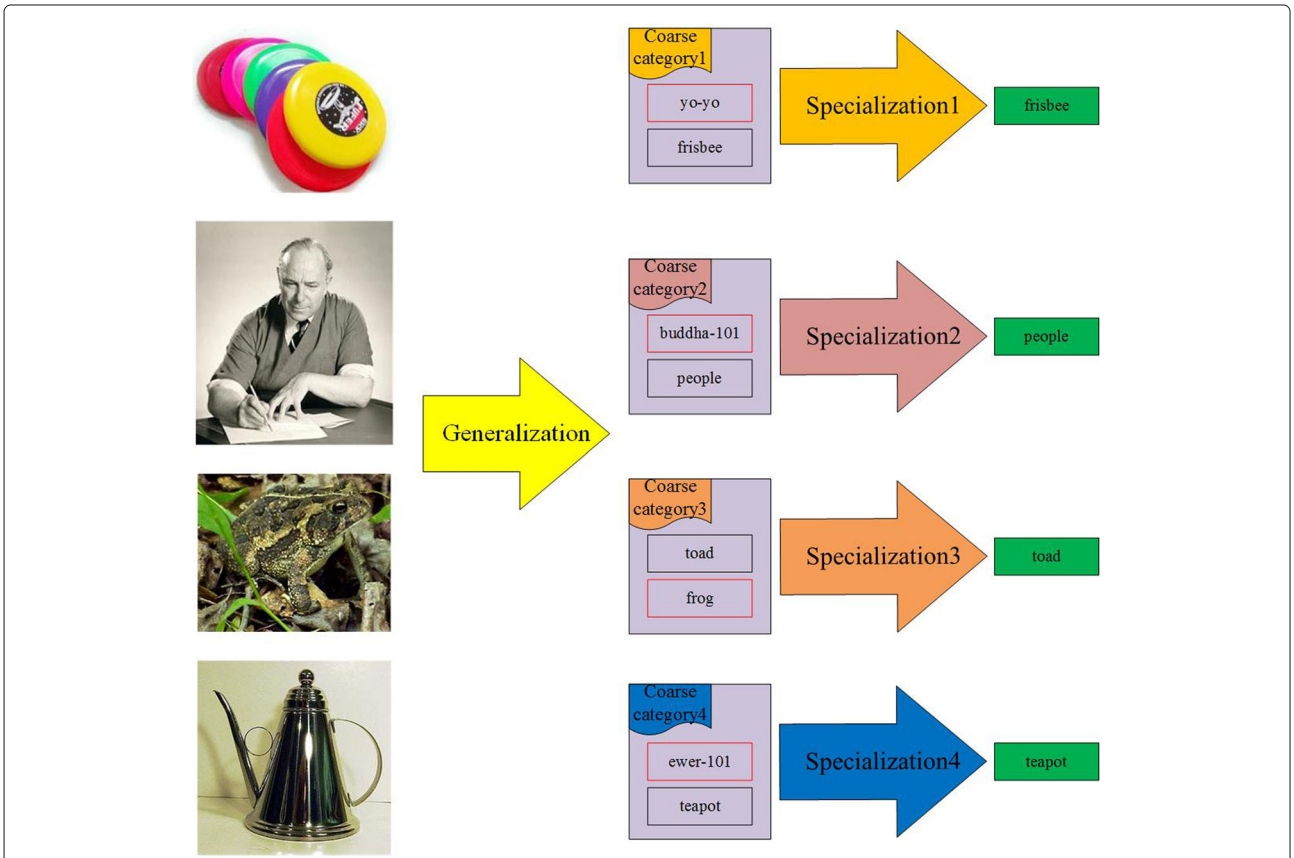

Fig. 9 Some images in Caltech256 were misclassified by the Trunk-CNN and corrected by Branch-CNNs. The red box in Coarse categories represents the output of the fine category by Trunk-CNN. The green box represents the final output of the Tree-CNN

and the galaxy are always misclassified to the people, the bowling-ball, the hourglass, and the light-house, respectively. Due to the Tree-CNN based on GoogLeNet, the accuracy of the Jesus Christ, the snail, the traffic-light, and the galaxy has been increased by 35.3, 26.1, 21.0, and 21.0%, respectively.

Figure 9 shows some Caltech256 test image dataset which are misclassified by the Trunk-CNN and corrected by Branch-CNNs. A pile of Frisbee, misclassified to a yo-yo, is presented in the first image. The misclassification is because the Frisbees are stacked together and a yo-yo is similar in appearance. The second image displays a man who is writing. The man is misclassified to Buddha because the background color around the man resembles the halo surrounding Buddha. The toad in the third image is misclassified to a frog, since the two only have a slight difference. The hind legs of a frog are longer than those of a toad. Besides, a frog's skin is moist and smooth, while that of a toad is dry and bumpy. The teapot in the fourth image is misclassified to a ewer. The two are differentiated from each other in that a teapot is used for making and serving tea while an ewer is used for carrying water.

The experiments show our method can improve the accuracy of classification on datasets Caltech101 and Caltech256. The main promotion on the categories with our method is often misclassified with the similar categories in original classifiers. The curve of accuracy of some categories, the category does not have similar category on dataset, has few fluctuation between our method and original method. Our method aims to improve the performance on medium dataset with some similar categories, and the results fit out theory.

4 Conclusions

The long-term goal of the CNN is to distinguish the objects with slight differences, as well as to differentiate one coarse category from another. But the current image dataset fails to collect categories with similar appearance. In addition, top 5 are preferred to be used as criteria in the current evaluation, but only the top 1 is useful in most conditions. It leads to current CNN fatigue to distinguish similar categories. We propose to divide an image dataset into smaller ones. The divided image dataset is comprised of categories in which objects share similar appearance. Therefore, the Tree-CNN is proposed. The main advantage of Tree-CNN is it significantly improved the accuracy of similar categories in the image dataset, while the computation in the training stage becomes heavier. The experiments of Caltech256 and Caltech101 demonstrate that the Tree-CNN outperform the original CNN clearly.

Abbreviations

CNN: Convolutional neural network

Acknowledgements

The authors would like to thank Tingfa Xu for the support.

Funding

This work was supported by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802.

Availability of data and materials

All data are fully available without restriction.

Authors' contributions

SJ and TX conceived of the Tree-CNN idea, and SJ was responsible for the programming. JG and JZ verified the analytical methods. SJ wrote the manuscript, and all authors revised the final manuscript. TX is the corresponding author. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 8 April 2018 Accepted: 3 July 2018
Published online: 04 September 2018

References

1. N Dalal, B Triggs, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference On*. Histograms of oriented gradients for human detection, vol. 1 (IEEE, 2005), pp. 886–893
2. DG Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
3. A Krizhevsky, I Sutskever, GE Hinton, in *Advances in Neural Information Processing Systems*. Imagenet classification with deep convolutional neural networks, (2012), pp. 1097–1105
4. C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, A Rabinovich, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Going deeper with convolutions, (2015), pp. 1–9
5. K He, X Zhang, S Ren, J Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Deep residual learning for image recognition, (2016), pp. 770–778
6. S Xie, R Girshick, P Dollár, Z Tu, K He, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Aggregated residual transformations for deep neural networks (IEEE, 2017), pp. 5987–5995
7. MD Zeiler, R Fergus, in *European Conference on Computer Vision*. Visualizing and understanding convolutional networks (Springer, 2014), pp. 818–833
8. J Goldberger, S Gordon, H Greenspan, Unsupervised image-set clustering using an information theoretic framework. IEEE Trans. Image Process. **15**(2), 449–458 (2006)
9. Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, R Girshick, S Guadarrama, T Darrell, Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
10. F Zhao, W Wang, H Chen, Q Zhang, Interference alignment and game-theoretic power allocation in MIMO heterogeneous sensor networks communications. Signal Process. **126**, 173–179 (2016)
11. F Zhao, X Sun, H Chen, R Bie, Outage performance of relay-assisted primary and secondary transmissions in cognitive relay networks. EURASIP J. Wirel. Commun. Netw. **2014**(1), 60 (2014)
12. F Zhao, B Li, H Chen, X Lv, Joint beamforming and power allocation for cognitive MIMO systems under imperfect CSI based on game theory. Wirel. Pers. Commun. **73**(3), 679–694 (2013)
13. F Zhao, H Nie, H Chen, Group buying spectrum auction algorithm for fractional frequency reuse cognitive cellular systems. Ad Hoc Netw. **58**, 239–246 (2017)
14. F Zhao, L Wei, H Chen, Optimal time allocation for wireless information and power transfer in wireless powered communication systems. IEEE Trans. Veh. Technol. **65**(3), 1830–1835 (2016)
15. K Simonyan, A Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

16. R Girshick, in *Proceedings of the IEEE International Conference on Computer Vision*. Fast R-CNN, (2015), pp. 1440–1448
17. R Girshick, J Donahue, T Darrell, J Malik, Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern. Anal. Mach. Intell. **38**(1), 142–158 (2016)
18. S Ren, K He, R Girshick, J Sun, in *Advances in neural information processing systems*. Faster R-CNN: towards real-time object detection with region proposal networks, (2015), pp. 91–99
19. J Redmon, S Divvala, R Girshick, A Farhadi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. You only look once: unified, real-time object detection, (2016), pp. 779–788
20. E Maggiori, Y Tarabalka, G Charpiat, P Alliez, Convolutional neural networks for large-scale remote-sensing image classification. IEEE Trans. Geosci. Remote Sens. **55**(2), 645–657 (2017)
21. F Hu, G-S Xia, J Hu, L Zhang, Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. **7**(11), 14680–14707 (2015)
22. G Antipov, S-A Berrani, J-L Dugelay, Minimalistic CNN-based ensemble model for gender prediction from face images. Pattern Recogn. Lett. **70**, 59–65 (2016)
23. G Antipov, M Baccouche, S-A Berrani, J-L Dugelay, Effective training of convolutional neural networks for face-based gender and age prediction. Pattern Recogn. **72**(C), 15–26 (2017)
24. J MacQueen, *et al*, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Some methods for classification and analysis of multivariate observations, vol. 1 (Oakland, 1967), pp. 281–297
25. T Zhang, R Ramakrishnan, M Livny, in *ACM Sigmod Record*. Birch: an efficient data clustering method for very large databases, vol. 25 (ACM, 1996), pp. 103–114
26. T-Y Lin, P Dollár, R Girshick, K He, B Hariharan, S Belongie, Feature pyramid networks for object detection. arXiv preprint arXiv:1612.03144 (2016)
27. P Kontschieder, M Fiterau, A Criminisi, S Rota Bulo, in *Proceedings of the IEEE International Conference on Computer Vision*. Deep neural decision forests, (2015), pp. 1467–1475
28. L Fei-Fei, R Fergus, P Perona, One-shot learning of object categories. IEEE Trans. Pattern. Anal. Mach. Intell. **28**(4), 594–611 (2006)
29. G Griffin, A Holub, P Perona, Caltech-256 object category dataset (2007)