Computers, Materials & Continua CMC, vol.62, no.2, pp.731-746, 2020

A Novel Combinational Convolutional Neural Network for Automatic Food-Ingredient Classification

Lili Pan[1], Cong Li[1,*], Samira Pouyanfar[2], Rongyu Chen[1] and Yan Zhou[1]

Abstract: With the development of deep learning and Convolutional Neural Networks (CNNs), the accuracy of automatic food recognition based on visual data have significantly improved. Some research studies have shown that the deeper the model is, the higher the accuracy is. However, very deep neural networks would be affected by the overfitting problem and also consume huge computing resources. In this paper, a new classification scheme is proposed for automatic food-ingredient recognition based on deep learning. We construct an up-to-date combinational convolutional neural network (CBNet) with a subnet merging technique. Firstly, two different neural networks are utilized for learning interested features. Then, a well-designed feature fusion component aggregates the features from subnetworks, further extracting richer and more precise features for image classification. In order to learn more complementary features, the corresponding fusion strategies are also proposed, including auxiliary classifiers and hyperparameters setting. Finally, CBNet based on the well-known VGGNet, ResNet and DenseNet is evaluated on a dataset including 41 major categories of food ingredients and 100 images for each category. Theoretical analysis and experimental results demonstrate that CBNet achieves promising accuracy for multi-class classification and improves the performance of convolutional neural networks.

Keywords: Food-ingredient recognition, multi-class classification, deep learning, convolutional neural network, feature fusion.

1 Introduction

People's demand for health has gradually escalated with the improvement of living standards. People pay more attention to the nutrition matching and food safety when choosing ingredients, which is becoming an essential factor for the improvements of life's quality. At present, food suppliers mainly rely on manpower for classification and quality identification of food ingredients. But it is a laborious task causing lots of cost on human, material, and management. Therefore, it is imperative to design an automatic food classification system.

Food recognition is an emerging topic in the field of computer vision, which is growing

[1] College of Computer Science and Information Technology, Central South University of Forestry & Technology, Changsha, 410114, China.

[2] School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA.

[*] Corresponding Author: Cong Li. Email: lcong1203@163.com.

rapidly. Martinel et al. [Martinel, Piciarelli and Micheloni (2016)] provided a system called Supervised Extreme Learning Committee (SELC), which extracted as many different features as possible but exploited only a subset of those for food classification. He et al. [He, Kong and Tan (2016)] proposed an automatic food classification approach, DietCam, using a texture verification model and a combination of a deformable part-based model for detecting food ingredients.

Recently, the research studies have resorted to the CNNs for image recognition. Besides, there have been plenty of achievements using deep learning to perform food image recognition that gained excellent performance. Kagaya et al. [Kagaya, Aizawa and Ogawa (2014)] applied CNNs to food detection, and found that CNNs can significantly enhance food detection. Liu et al. [Liu, Cao, Luo et al. (2016)] proposed a new algorithm to analyze the food images based on CNNs, and achieved remarkable performance on two public food image datasets (UEC-256 and Food-101). However, the existing food identification methods mainly focused on food dishes and there are few literatures on the classification of food ingredients. A few food-ingredient datasets are available, but the scale is not enough to train a deep model which requires large datasets to avoid overfitting. Fang et al. [Fang, Zhang, Sheng et al. (2018)] combined Deep Convolutional Generative Adversarial Networks (DCGAN) with CNN. The DCGAN is utilized to generate food samples to be further used in the training of the image recognition model, but training GAN was so difficult.

One of the main challenges is that how to get a high-performance deep learning model on the limited training samples. Till now, there are two widely-used methods worked well. The first solution is using fine-tuning technology that takes a pre-trained model in which the trained weights are used as the initialization weights of a specific task, and then resume the training. [Singla, Yuan and Ebrahimi (2016)] fine-tuned GoogLeNet with only updating the parameters of the last six layers, reported the accuracy of 83.6% on food-11. Another approach is to adjust the network architecture appropriately based on the basic networks, to improve the framework's performance. In the work Hou et al. [Hou, Liu and Wang (2017)], a framework called DualNet was developed that coordinated two parallel CNNs to learn complementary features. The DualNet performed better than the baselines on UEC FOOD-100, but the highest accuracy was only about 50%. Many existing schemes simply applied the classical CNN models and ignored the fact that the corresponding architecture should be designed for a specific classification task. And it is difficult for a single network to fully understand the details of the image.

In order to solve the aforementioned problems, this paper presents a new combinational convolutional neural network for multi-class classification of food ingredients. In this work, a new combinational network is constructed which combines two convolutional subnets in parallel. Firstly, the complexity of the networks is optimized with adjusting and modifying the existing convolutional networks. Then the feature fusion method is used to coordinate the training of subnets to complement and improve the learned information, which makes the extracted image features more efficient and accurate. The detailed experimental results demonstrate that the novel food-ingredient classification framework based on combinational networks can effectively improve the generalization ability of the model and obtains excellent recognition accuracy.

The organization of this paper is as follows. In Section 2, a brief description of the advanced techniques in food detection and CNNs is introduced. We present the details of the proposed CBNet in Section 3. Section 4 analyzes the experimental results on CBNets and corresponding subnets. Finally, the paper is concluded in Section 5.

2 Related work

This section briefly describes the existing research in food detection and CNNs.

2.1 Food detection

Recently, there are several progresses regarding the food classification in the literature. Chen et al. [Chen, Yang, Ho et al. (2012)] introduced a primary method for the quantity estimation based on depth information to deal with the issues of feature descriptors in the food identification. Using this method, they achieved accuracy of 68.3% on a dataset including 50 major categories of worldwide food and 100 images for each category. Li et al. [Li, Qin, Xiang et al. (2018)] addressed the shortcomings of Harris corner detection, and proposed a Harris feature point selection algorithm based on adaptive threshold. [Farinella, Moltisanti and Battiato (2014)] utilized visual words distributions (Bag of Textons) for representing food images, and Support Vector Machine as the classifier. The experimental results of applying the proposed method to Pittsburgh Fast-Food Image Dataset have showed promising performance. Bossard et al. [Bossard, Guillaumin and Van Gool (2014)] presented a novel method used for mining discriminative visual components and efficient classification based on Random Forests.

In recent years, deep learning has been utilized to construct abundant state-of-the-art methods on food recognition. Ciocca et al. [Ciocca, Napoletano and Schettini (2017)] designed a suitable automatic tray analysis pipeline to find the regions of interest, and predicted the corresponding food class for each region. Compared to other visual descriptors, the CNNs-based features obtained the best performance. CNNs have become the dominant machine learning approach for visual object recognition. The research studies illustrated the remarkable accomplishment of the convolutional networks. At present, CNN has been widely used in food image recognition and achieved excellent performance. Hassannejad et al. [Hassannejad, Matrella, Ciampolini et al. (2016)] fine-tuned the GoogLeNet to deal with the food classification problem.

Transfer Learning is a machine Learning method that transfers knowledge in one field (source field) to another field (target field) and could achieve better learning results in the target field. Pan et al. [Pan, Pouyanfar, Chen et al. (2017)] proposed a new framework called Deepfood, extracted rich and effective features from food-ingredient images using ResNet model. They improved the average classification accuracy by applying Information Gain algorithm and Sequential Minimal Optimization (SMO) classifier.

Properly adjusting the architecture of classical networks is also an effective method. Martinel et al. [Martinel, Foresti and Micheloni (2018)] introduced a new deep scheme that was designed to handle the food structure. Outputs of the deep residual blocks were combined with the sliced convolution to produce the classification score for specific food categories. Experimental results demonstrated that the provided solution showed better

performance compared to the existing approaches (e.g., a top-1 accuracy of 90.27% on the Food-101 challenging dataset). Chen et al. [Chen, Xu, Xiao et al. (2018)] proposed a fast auto-clean CNN model for online prediction of food materials. It focused on the complex characteristics of the food images such as the complexity of the food materials, the dislocation and the uniformity of illumination. The experimental results illustrated that the methodology enhanced the efficiency and accuracy for food classification.

2.2 Convolutional neural networks

Deep learning is a machine learning method which is derived from the research of artificial neural networks, aiming to build a neural network that could simulate human brain to analyze and understand data. Currently in the field of image processing, deep convolutional neural network has made a series of breakthroughs.

VGGNet [Simonyan and Zisserman (2014)] is a deep convolutional neural network proposed in the Visual Geometry Group (VGG) team of Oxford University in 2015. It mainly discussed the importance of deep neural networks for training, and built a variety of network structures, among which the 19-layer deep network had achieved the best performance. It inherited some of the frameworks of LeNet and AlexNet, and gained excellent performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014.

GoogLeNet [Szegedy, Liu, Jia et al. (2015)] is a deep learning structure in 2014. The whole network structure was connected by several Inception modules. The Inception module changed the complete connection of the convolution layer into a sparse connection. By using the filter of inconsistent size, the features were extracted on multiple scales. The highly correlated features were connected together, while the irrelevant noncritical features were weakened. Therefore, the redundant information of the features output by the Inception module was less which could accelerate the training convergence.

The Deep Residual Network (ResNet) [He, Zhang, Ren et al. (2016)] is a deep learning network proposed by Kaiming He. In 2015, it ranked first in ImageNet image classification, object positioning, COCO object detection and COCO image segmentation. Its important achievement was reconstructing the learning process, which redirected the deep neural network information flow and successfully solved the accuracy saturation and vanishing gradients in deep neural networks.

DenseNet [Huang, Liu, Van Der Maaten et al. (2017)] gained the best paper of CVPR (Computer Vision and Pattern Recognition) 2017, which is a convolutional neural network with dense connections. It connects each layer to every other layer utilizing a feed-forward fashion. Accordingly, the feature maps of all previous layers are used as the inputs for each layer, and the feature vectors of current layer are also used as inputs for all subsequent layers. The approach not only alleviated the problem of vanishing-gradient, strengthened feature propagation, and encouraged feature reuse, but also substantially reduced the number of parameters.

3 The combinational convolutional neural network (CBNet)

In this section, we will describe the details of the proposed CBNet. Fig. 1 shows the

architecture of CBNet. When training on food-ingredient datasets, CBNet receives the feedback of classification from the auxiliary classifiers and integrates the results using the fusion classifier which is placed in the final part of the framework. Then CBNet guides and regulates the learning direction of the subnets. Therefore, the information learned with two subnets would be more complementary and discriminative. When these complementary features (form as feature maps) are received, the feature fusion component will further purify them. Finally, CBNet obtains the features which are highly related to the food ingredients for classification and accurately outputs the predicted categories. This component simplifies the subnets and taking a modified Inception module as a feature fusion component, which are indeed the linchpin of CBNet. In the following subsections, we will illustrate the process of subnets simplification and network combination in details.

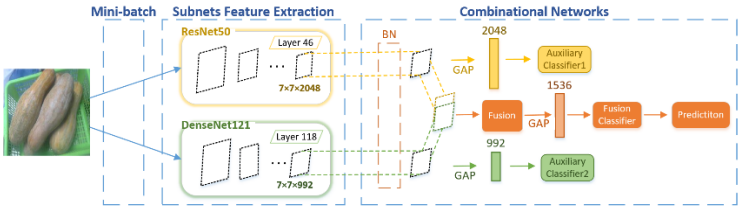

Figure 1: The CBNet framework for food classification

3.1 Simplify subnets

In order to resolve the complex tasks as much as possible, the fitting ability (number of parameters) of the machine learning algorithm is usually higher than the complexity of the problem. This leads to the overfitting problem. The concrete manifestation of overfitting is that the final model has a good performance on training set, but poor on test set. In other words, the generalization ability of the model is weak. Particularly, the overfitting can easily happen on a complex network when the training data is limited. Due to the weak scale of food-ingredient dataset, we decrease the layers of subnets to reduce the complexity of the network. As shown in Fig. 2, CNNs are mostly stacked by the same structure, such as the Residual block of ResNet and the Dense block of DenseNet. A simple strategy is to decrease the modules after the last 2×2 average pooling layer step by step.

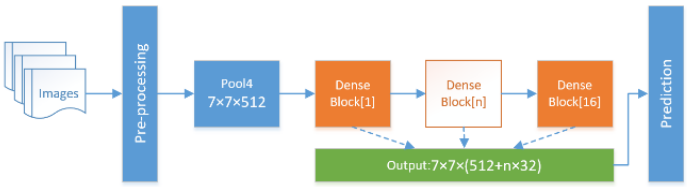

Figure 2: The simplification of DenseNet121

Taking DenseNet121 as an example, the size of feature map is 7×7 with 512 channels (denoted as $7 \times 7 \times 512$) when the input image reaches to the layer pool4. The

subsequent network is composed of 16 Dense blocks. Once it goes through a block, the size of feature map does not change, but the number of channels increases by 32. It can be seen that the final output shape is $7 \times 7 \times (512 + n \times 32)$. We gradually reduce the number of blocks from the deepest layer, then evaluate the Feature Map output of each block. Finally, the best Feature Map output is determined.

Algorithm 1. Subnet Selection Algorithm

Input: *base_model* and the *dataset*
Output: the *best_model*, *best_Acc*, and the number of blocks *k*
for $Block_i$ do:
  model = Load (*base_model*)
  model = Model (inputs=*image*,ouputs = $Block_i.output$)
  Classifier = Dense (*classes*, softmax) [GAP (output)]
  new_model = *Input*→*model*→ *Classifier*
  new_model.Fit (Training, Validation, checkpoint = val_loss)
  Acc_i = *new_model*.evaluate (Evaluation)
end for
best_Acc = Max (*Acc*)
k = *Acc*.Index (*best_Acc*)
best_model = Model (inputs = *image*, ouputs = $Block_k.output$)
Return *k*, *best_Acc* and *best_model*

The subnet selection algorithm is shown in Algorithm 1. the input of the algorithm includes the subnet models selected from the classical CNNs and the food-ingredient dataset. Its output is the best simplified model, the corresponding accuracy and the number of modules it contains. After loading the base model, the algorithm evaluates the accuracy of the feature map output extracted from each block on food-ingredient dataset. In the first loop, different numbers of blocks are connected after the last 2×2 pooling layer. Then, we add the GAP (Global Average Pooling) layer, which is connected to the classifier and thus a wider network is constructed. In the next loop, we start to train the model and set a checkpoint to monitor the changes of validation loss. The trained model is used to evaluate the accuracy of the testing set. Through comparing and analyzing the models that contain different modules, Algorithm 1 finally returns the best model, the corresponding accuracy and the number of blocks in the selected model.

Particularly for VGG19, the fully connected layer accounts for nearly 90% of parameters which easily results in overfitting. Using the concept of global average pooling proposed by NIN (Network in Network) [Lin, Chen and Yan (2014)], this framework replaces the fully connected layer with the global average pooling layer. The experiments prove that the novel model trained faster and could effectively reduce overfitting.

3.2 Feature fusion component

Inspired by GoogLeNet, a new Inception module is designed as the feature fusion component of the CBNet displayed in Fig. 3. It is used to fuse different features, filter the redundant information, and collect important features for image classification.

Compared with the original modules, InceptionV3 [Szegedy, Vanhoucke, Ioffe et al. (2016)] used the asymmetric convolution splitting which can process more and richer

spatial features. In this paper, the feature fusion component is designed that redistributes the number of convolutional cores of different size in InceptionV3 module. It appropriately improves the proportion of 1×1 convolution kernels at the Avgpool branch, also the number of channels is increased from 192 to 336, so the ability of subnets classification is retained as much as possible. The input images are respectively convolved with different convolution kernels in four branches to obtain different receptive fields and maintain the diversity. Meanwhile, 1×1 convolution kernel's capacity of organizing information across channels could facilitate feature fusion, as well as removing computational bottlenecks and controlling dimensions. Thus, the output dimension of the fused feature is fixed at 1536.

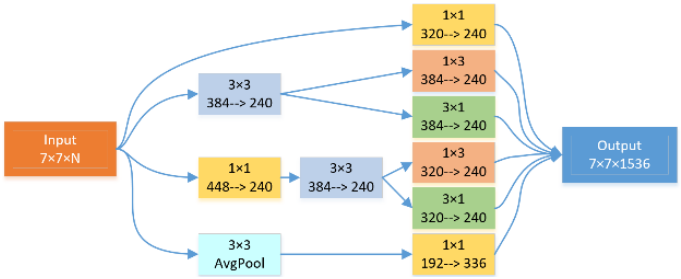

Figure 3: Feature fusion component

3.3 Fusion strategy

The features aggregated by two-stream of subnets are more diverse compared to the single-stream CNN, which is conducive to train a deep learning model with stronger discrimination ability. How to ensure the diversity and prevent the subnet interference are the key factors of the CBNet. Therefore, a new fusion strategy is proposed in this part, which is helpful to improve the collaboration and complementarity between the two subnets.

3.3.1 BN (Batch Normalization)

Batch Normalization [Ioffe and Szegedy (2015)] is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance. It is common to utilized the BN transform to manipulate any activation in networks. Considering a mini-batch B of size m, given one layer's inputs $X = \{x_1, x_2, \ldots, x_m\}$, the activations $Y = \{y_1, y_2, \ldots, y_m\}$ can be calculated as Eq. (1).

$$y_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{1}$$

where μ_B and σ_B^2 are the mini-batch mean and variance respectively. For each activation x_i, there are a pair of parameters γ, β, which scale and shift the normalized value. The constant ϵ is added to the mini-batch variance for numerical stability. In this work, BN is placed before the convergence of the two streams for eliminating the different distribution of the subnets. This method avoids one stream to be ignored in the procedure of fusion. In the meantime, the setting of BN layer also can contribute to accelerate convergence and control the overfitting.

3.3.2 Auxiliary classifier

The CBNet uses auxiliary classifier to make the output of subnets as the classification scores. The signal of backpropagation increases in the training process, which lets the subnet possess the capacity to learn independently and classify well. With the additional classifiers, CBNets can perceive more details and learn about the situation of subnets training with the following loss function:

$$L = \lambda_F L_F + \lambda_{S1} L_{S1} + \lambda_{S2} L_{S2} \quad (2)$$

where L_F , L_{S1} , L_{S2} are cross entropy loss computed with the Softmax_Fusion, Softmax_S1, and Softmax_S2, respectively. The loss weight vector empirically is set as $[1,0.3,0.3]$. At the time of global fine-tuning the whole network, CBNet can adjust the learning direction timely once that the subnets deviate from the normal range. So that subnets can cooperate fine and fully learn the complementary features. In addition, the auxiliary classifier can speed up the convergence of the deep network.

Algorithm 2. Subnet Combination Algorithm

Input: Two subnets and the corresponding output layers L_1, L_2
Output: Combinational model *CBNet*
1: x_1 = *subnet1.*L_1*.output* → BN
2: x_2 = *subnet2.*L_2*.output* → BN
3: x = Concate ($[x_1, x_2, axis = 3]$) →Fusion Component
4: *Classifier* = Dense (*classes*, softmax, loss_weight = $[1.0, 0.3, 0.3]$)
5: *model* = Model (inputs = [*subnet1.input, subnet2.input*], ouputs = $[x, x_1, x_2]$)
6: *CBNet* = *model* → *Classifier*
7: Return *CBNet*

The subnet merging technique is descripted in Algorithm 2. The input of the algorithm is the model of two subnets with their corresponding output layers, and the output of the algorithm is CBNet combined with the subnets. The procedure of the algorithm includes 3 steps. Step 1: Model combination algorithm extracts the features of subnets and carries out the BN operation, expressing as x_1 (*height*, *width*, *channels1*) and x_2 (*height*, *width*, *channels2*) respectively. Step 2: The subnets' feature maps are concatenated on the channels. Then, the feature fusion component collects information on the combined feature maps with different convolution kernels to obtain the fusion features x (*height*, *width*, *1536*) in which the dimension is fixed at 1536. Step 3: x_1, x_2, and x are connected in the classifier whose output has been modified as the categories of the corresponding dataset. Ultimately the complementary constraint is imposed by weighting to construct the CBNet.

4 Experimental analysis

In this section, we evaluate CBNet and compared it with the well-known deep learning models such as VGGNet, ResNet and DenseNet. In order to prove the effectiveness and efficiency of the CBNet, the last layer of the networks is fine-tuned in the Scheme 1 (Fine-tuning the Last Layer) as shown in Section 4.3. In the second scheme (Fine-tuning the Whole Network), all layers of the networks are fine-tuned to learn the features which are highly relevant to the target dataset. Overfitting easily occurs with insufficient training samples and complex models. Therefore, this paper attempts to reduce the

number of subnet layers in the third scheme (Simplifying CBNet), to improve the complexity of networks.

4.1 Datasets and environments

A food-ingredient dataset named Food-41 set has been created for food-ingredient recognition. The dataset of food ingredients is collected from a large food supply chain platform called Mealcome (MLC dataset) [Chen, Xu, Xiao et al. (2018)] in China. The noisy pictures were removed and clearly distinguished pictures were picked. In addition, there are a few images supplemented by manual collection. Then labeled them with different food types. Finally, the dataset contains 4100 images and 41 classes, which are divided into three parts, 60% for training set, 20% for validating set and 20% for testing set. So as to achieve better performance, we resize the images to a size of 640×480 px.

All the networks are implemented with a popular Deep Learning framework called Keras and trained/tested on a high-end server with 16 GB of RAM equipped with a NVIDIA GTX1070 with 8 GB of memory and 1920 CUDA cores.

4.2 Experiments setting

According to the Algorithm 1, we build different CBNets based on the classical network VGG19, ResNet50 and DenseNet121, then validating its performance on Food-41 dataset. Each model was previously trained on ImageNet. The output of the models has been changed to the number of classes for the dataset.

The proposed method is evaluated using the following three settings:

(1) Fine-tuning the last layer to show the effectiveness of the CBNet in a short time. Particularly, for VGG19 (VGG19-Dense), we replace the original two fully connected layers with a GAP layer (VGG19-AVG) following the method mentioned in Section 3.1. (2) Fine-tuning fewer network parameters is a compromise on computing resources and time. In Scheme 2, the whole network is fine-tuned to achieve a higher accuracy. Besides, since the linchpin of CBNet is to coordinate two subnets to learn complementary features from input images, the learning processes of the networks are visualized, and then the collaboration and complementarity of the CBNet are analyzed and verified in detail. (3) As mentioned in Section 3.1, the overfitting is inevitable when a complex network with less data is trained. Therefore, the number of layers in the subnets are reduced in Scheme 3 so that a better output feature map is obtained and the CBNet is also optimized.

The image size (network input) is $224 \times 224 \times 3$, without applying a data enhancement. When evaluating the accuracy on testing set, the model which gains the high-performance loss value on validation set is chosen. We take the average accuracy ultimately through three different experiment schemes. Hyper-parameter adjustment is also an important part, SGD is selected as the optimizer and used momentum with a decay of 0.9. The appropriate learning rate is set for each network, decaying every two epochs using an exponential rate of 0.94. The mini batch size is set to 16, training 30 epochs totally.

4.3 Performance evaluation and analysis

4.3.1 Scheme 1: Fine-tuning the last layer

In this experiment, only the last layer softmax classifier is updated to verify the effectiveness of CBNet in a short time. Performance results on VGG19-Dense, VGG19-AVG, ResNet50, DenseNet121 and the derived CBNet are shown in Tab. 1. It illustrates the training time, feature dimension and accuracy of each networks.

Table 1: The accuracy of fine-tuning the last layer

Model	Time (s/epoch)	Feature Dim.	Accuracy (%)
VGG19-Dense	27	4096	86.02
VGG19-AVG	19	512	88.82
ResNet50	19	2048	84.14
DenseNet121	20	1024	83.37
CBNet-VR	45	1536	88.90
CBNet-VD	42	1536	89.47
CBNet-RD	40	1536	88.33

We can see that VGG19-AVG has attains the highest average accuracy among all the single models, while requiring less time and lower feature dimension. The conception of CBNet is to coordinate the parallel learning of two subnets and extracts complementary features for classification. It is appropriate to use the highest accuracy of the single models as the baseline. By comparing with the single models, CBNets have a certain improvement generally, in which CBNet-VD attains the best performance and the accuracy reaches up to 89.47%.

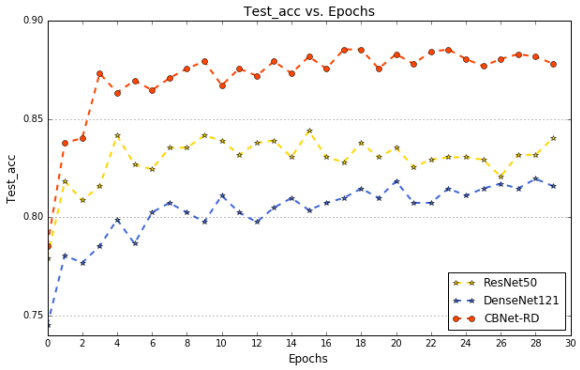

Figure 4: Accuracy comparison between CBNet and single networks

It is worth mentioning that comparing with two single models [ResNet50, Dense-Net121], the accuracy of CBNet-RD is as high as 88.33%, increased by 4.19%. Fig. 4 shows the accuracy comparison between CBNet-RD and single networks, when the number of epochs increases, the test accuracy becomes higher and more stable. Moreover, the CBNet-RD converges faster, and the accuracy reaches to over 85% in the third epoch, which is higher than the highest accuracy of the single models. It is concluded that the

feature fusion method can effectively integrate the features of two subnets and finally extracts more relevant deep features for the classification.

4.3.2 Scheme 2: Fine-tuning the whole network

The CBNet is evaluated with fine-tuning all the layers of the network in this experiment, and VGG19-Dense is ignored because VGG19-AVG shows the better performance.

Table 2: The accuracy of fine-tuning the whole network

Model	Time (s/epoch)	Accuracy (%)	*vs.* Scheme 1
VGG19-AVG	53	92.36	+3.54
ResNet50	41	93.29	+9.15
DenseNet121	43	93.78	+10.14
CBNet-VR	91	94.03	+5.13
CBNet-VD	93	95.00	+5.53
CBNet-RD	88	95.28	+6.95

Fine-tuning overall network is more advantageous as shown in Tab. 2. In all the single models, DenseNet121 becomes the optimal model, where the accuracy reaches to 93.78%, raising by 10.41% compared to the Scheme 1. On the contrary, it is the lowest accurate rate in Experiment 1. In order to further understand the relationship between retrained layers of DenseNet and the learning effect, the feature maps of different sizes are visualized in the forward propagation as shown in Fig. 5.

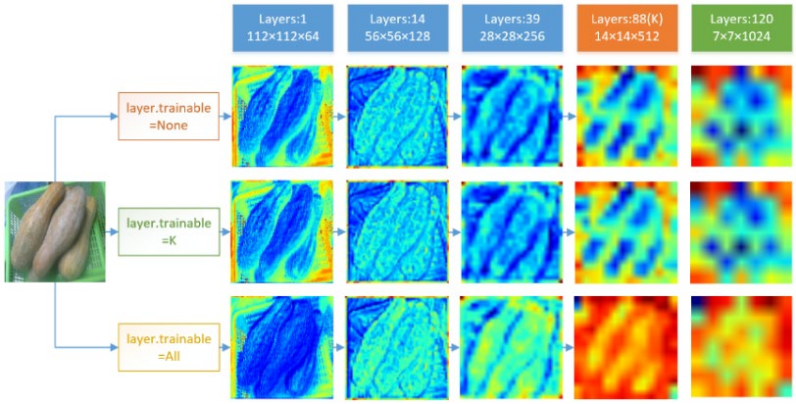

Figure 5: The learning process of DenseNet121 from shallow to deep

Fig. 5 shows the learning process of DenseNet121 from shallow to deep. The first branch is the pre-trained model without fine-tuning, whose weights are trained on the large dataset ImageNet. The visualization results of each layer show the hierarchical characteristics of the network. It is observed that the shallow convolutional layers learned some basic features such as color, shape, contour and so on. As we go deeper through the network layers, the 'noise' is gradually removed and the features contains more comprehensive information.

Compared with branch 1, there is almost no distinction on branch 2. This is because that branch 2 is only retrained the convolution layer after 88 layers and froze the shallow. The method limits the ability of the network to learn the deep abstract features. The third branch fine-tunes the overall network, which gives the network the ability to learn the features related to the specific classification task. Accordingly, the shallow-layer features show the main body of food ingredients better than the previous two branches. It can be seen from Fig. 5 that the retrained network weakens the activation degree of the background intelligently in layer39. The layer120 learned the key discriminative features, which is conducive to the classification on the dataset. The highest accuracy of the single models is taken as the baseline. CBNet still performs better in Scheme 2, which proves the effectiveness of the method we proposed. Compared to the Scheme 1, CBNet-RD becomes better than CBNet-VD for food recognition. Its accuracy reaches to 93.78%, which is 1.22% higher than the single models [VGG19, DenseNet121].

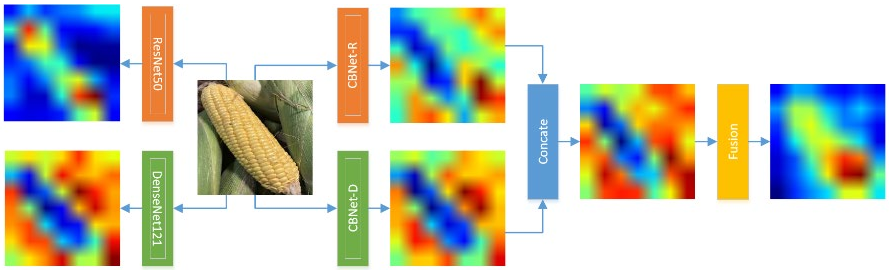

Figure 6: The differences of visual feature maps

Fig. 6 shows the differences of visual feature maps between single networks and the subnets. Comparing to the independent trained ResNet model, the features extracted by subnets have an obvious distinction, it indicates that the learning direction of subnet has deviated from the initial direction. The phenomenon 'deviation' is a key aspect of the CBNet. CBNet aims to extract more features related to the food ingredients. In the light of the feedback of classification effect obtained by the additional auxiliary and the fused classifiers, the learning direction of the subnets is guided and adjusted in a timely manner. Thus, subnets can interpret pictures from distinct angles and learn more complementary features. After further refining with the fusion component, the obtained features can express the image more accurately.

4.3.3 Scheme 3: Fine-tuning the simplified CBNet

Referring to Algorithm 2, the number of blocks is cut down gradually from the deepest part of the subnets in this experiment. The performance is evaluated on CBNet-RD because it achieved the highest accuracy in Scheme 2. It is concluded that the novel CBNet-RD performs better when ResNet50 and DenseNet121 both drop one block. Denoting the simplify net by adding '-R' at the end of model name.

The performance of simplified combinational network CBNet-RD-R and the corresponding subnets are shown in Tab. 3. On the basis of Scheme 2, the accuracy of the simplified subnets is increased by 0.77 and 1.06, respectively. The performance of

CBNet-RD-R achieves 95.41%, which is 1.63% higher than the original single model (DenseNet121). From this experiment, it can be referred that applying the simplified strategy can effectively alleviate overfitting when the training sample is limited.

Table 3: The accuracy of fine-tuning the simplify networks

Model	Accuracy (%)	*vs.* Scheme 1
ResNet50-R	94.06	+0.77
DenseNet121-R	94.84	+1.06
CBNet-RD-R	95.41	+0.13

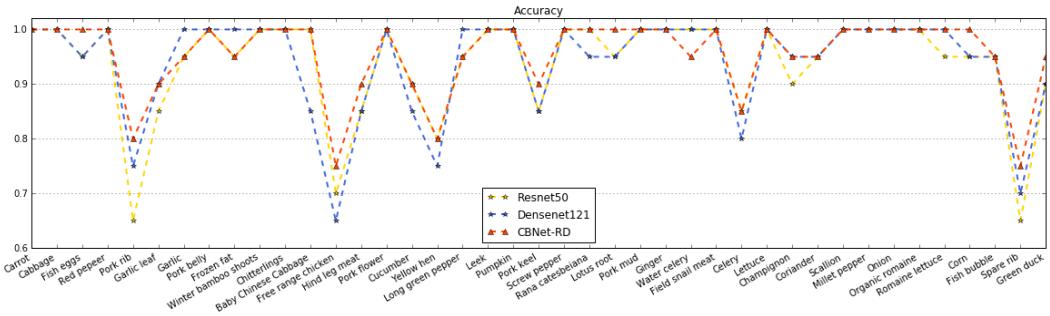

Figure 7: Accuracy for different models on various food ingredients

Fig. 7 shows the visualized performance comparison of different models evaluate on Food-41 dataset. It can be seen the proposed model CBNet-RD-R has an overall advantage in comparison with the single models. Observing the trend, the performance of several categories is not ideal on all models, but CBNet-RD-R's performance is obviously higher than other models, for instance, the recognition accuracy of Pork rib increased by 10%. Overall, the top-1 accuracy plot of CBNet-RD-R on almost all food-ingredient classes is fluctuating in the range of 90% to 100%, only a few under 90%.

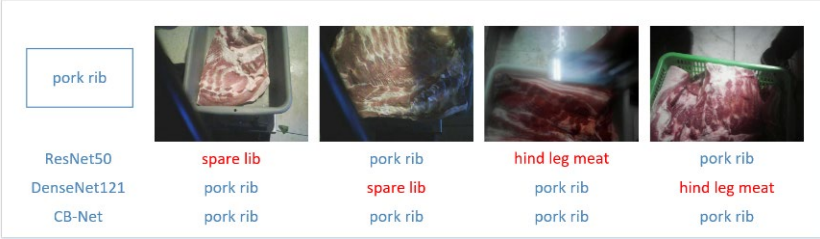

Figure 8: Examples of misclassified samples from single model classifiers results

By analyzing the performance of models on poor categories, it can be seen the main reason resulting in wrong prediction may be due to the fact that several categories are too similar. Some of the Pork rib images were wrongly predicted as Hind leg meat or Spare rib. It's also a bit difficult to distinguish category Free range chicken from Yellow hen, and Celery from Water celery by eyes. Fig. 8 shows the wrong prediction of Pork rib by single models. For these 4 images, only one of the single models could predict the right

category. Although these three categories are indeed extremely similar, the CBNet-RD-R has learned the complementary features by combination of two subnets, and all four images are correctly predicted using CBNet. The CBNet shows its superior higher recognition performance compared to other CNNs.

Table 4: Accuracy comparison with existing methods

Framework	Accuracy (%)	Model
Fine-tuning the Classifier	84.14	Pre-trained ResNet
Deepfood [Pan]	87.78	Pre-trained ResNet
CBNet-VD	89.47	Pre-trained V&D
CBNet-RD-R	95.41	Retrained

As can be seen from Tab. 4, compared to the basic Fine-tuning method, the 'Deepfood' in Pan et al. [Pan, Pouyanfar, Chen et al. (2017)] had a certain improvement using pre-trained ResNet model. The accuracy of CBNet-VD is higher than other deep learning benchmarks without changing the features of the original pre-training model. More importantly, the method in this paper achieves the best average accuracy using retrained CBNet-RD-R model, and the average accuracy achieves 95.41% which is a significant improvement comparing to the method in Chen et al. [Chen, Xu, Xiao et al. (2018)] and superior to the approach Deepfood. Besides the effect of fusion network, the promotion might due to optimization on dataset.

In conclusion, the extensive experimental results prove that the proposed framework has high performance for the multi-category classification of food ingredients. The new method can quickly and effectively combine two different convolutional networks, and improve the generalization performance of the network. This achieves a deep learning model with stronger discrimination ability under limited training samples, which significantly enhances the recognition accuracy.

5 Conclusion

In this work, a new automatic classification method is proposed for medium/small size datasets based on the CBNet which combines two different CNNs and extracts complementary features using deep learning. The novel framework contributes to alleviate the overfitting problem that is produced by almost all large networks owing to the limit number of training samples. The customized feature fusion method and the corresponding fusion strategy are the key aspects of CBNet. The feature fusion component can effectively eliminate redundancy and extract efficient and thriving features related to the images. Meanwhile, the corresponding fusion strategy can timely guide and adjust the learning direction of the subnets, so as to make subnets cooperate well with each other. Finally, CBNet based on VGGNet, ResNet and DenseNet is thoroughly investigated and evaluated on Food-41 dataset, which achieves higher top-1 accuracy than the baselines. We believe that other classification problems for small or medium datasets can benefit from CBNet which trains an excellent model with better generalization performance. For future work, we make an arrangement to survey the utilization of more advanced fusion strategy and simplification methods to further

optimize CBNet.

Acknowledgement: This paper is partially supported by National Natural Foundation of China (Grant No. 61772561), the Key Research & Development Plan of Hunan Province (Grant No. 2018NK2012), Postgraduate Research and Innovative Project of Central South University of Forestry and Technology (Grant No. 20183012), Graduate Education and Teaching Reform Project of Central South University of Forestry and Technology (Grant No. 2018JG005), and Teaching Reform Project of Central South University of Forestry and Technology (Grant No. 20180682).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bossard, L.; Guillaumin, M.; Van Gool, L. (2014): Food-101-mining discriminative components with random forests. *European Conference on Computer Vision*, vol. 8694, pp. 446-461.

Chen, H.; Xu, J.; Xiao, G.; Wu, Q.; Zhang, S. (2018): Fast auto-clean CNN model for online prediction of food materials. *Journal of Parallel and Distributed Computing*, vol. 117, pp. 218-227.

Chen, M. Y.; Yang, Y. H.; Ho, C. J.; Wang, S. H.; Liu, S. M. et al. (2012): Automatic chinese food identification and quantity estimation. *Special Interest Group on Computer GRAPHics and Interactive Techniques Asia 2012 Technical Briefs.*

Ciocca, G.; Napoletano, P.; Schettini, R. (2017): Food recognition: a new dataset, experiments, and results. *IEEE journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 588-598.

Fang, W.; Zhang, F.; Sheng, V. S.; Ding, Y. (2018): A method for improving CNN-based image recognition using DCGAN. *Computers, Materials & Continua*, vol. 57, no. 1, pp. 167-178.

Farinella, G. M.; Moltisanti, M.; Battiato, S. (2014): Classifying food images represented as Bag of Textons. *IEEE International Conference on Image Processing*, pp. 5212-5216.

Hassannejad, H.; Matrella, G.; Ciampolini, P.; De Munari, I.; Mordonini, M. et al. (2016): Food image recognition using very deep convolutional networks. *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, pp. 41-49.

He, H.; Kong, F.; Tan, J. (2016): Dietcam: multiview food recognition using a multikernel SVM. *IEEE journal of Biomedical and Health Informatics*, vol. 20, no. 3, pp. 848-855.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Hou, S.; Liu, X.; Wang, Z. (2017): Dualnet: learn complementary features for image recognition. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 502-510.

Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. (2017): Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700-4708.

Ioffe, S.; Szegedy, C. (2015): Batch normalization: accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, pp. 448-456.

Kagaya, H.; Aizawa, K.; Ogawa, M. (2014): Food detection and recognition using convolutional neural network. *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 1085-1088.

Li, H.; Qin, J.; Xiang, X.; Pan, L.; Ma, W. et al. (2018): An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE Access*, vol. 6, pp. 66963-66971.

Lin, M.; Chen, Q.; Yan, S. (2014): Network in network. *International Conference on Learning Representations*.

Liu, C.; Cao, Y.; Luo, Y.; Chen, G.; Vokkarane, V. et al. (2016): Deepfood: deep learning-based food image recognition for computer-aided dietary assessment. *Inclusive Smart Cities and Digital Health*, vol. 9677, pp. 37-48.

Martinel, N.; Foresti, G. L.; Micheloni, C. (2018): Wide-slice residual networks for food recognition. *IEEE Winter Conference on Applications of Computer Vision*, pp. 567-576.

Martinel, N.; Piciarelli, C.; Micheloni, C. (2016): A supervised extreme learning committee for food recognition. *Computer Vision and Image Understanding*, vol. 148, pp. 67-86.

Pan, L.; Pouyanfar, S.; Chen, H.; Qin, J.; Chen, S. C. (2017): Deepfood: automatic multi-class classification of food ingredients using deep learning. *IEEE 3rd International Conference on Collaboration and Internet Computing*, pp. 181-189.

Simonyan, K.; Zisserman, A. (2015): Very deep convolutional networks for large-scale image recognition. *In International Conference on Learning Representations*.

Singla, A.; Yuan, L.; Ebrahimi, T. (2016): Food/non-food image classification and food categorization using pre-trained Googlenet model. *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, pp. 3-11.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. (2016): Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826.