Pattern Recognition 140 (2023) 109550

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Learn from each other to Classify better: Cross-layer mutual attention learning for fine-grained visual classification☆☆

Dichao Liu[a,c,*], Longjiao Zhao[a], Yu Wang[b,1], Jien Kato[b]

[a] Graduate School of Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-Shi, Aichi-Ken, 464-8601, Japan
[b] College of Information Science and Engineering, Ritsumeikan University, 1, Nojihigashi, Kusatsu-Shi, Shiga-Ken, 525-0058, Japan
[c] Research Team, Navier, Inc., 9-2 Nibancho, Chiyoda-ku, Tokyo, 102-0084, Japan

ARTICLE INFO

Article history:
Received 6 June 2022
Revised 21 February 2023
Accepted 19 March 2023
Available online 22 March 2023

Keywords:
Fine-grained recognition
Image classification
Deep features

ABSTRACT

Fine-grained visual classification (FGVC) is valuable yet challenging. The difficulty of FGVC mainly lies in its intrinsic inter-class similarity, intra-class variation, and limited training data. Moreover, with the popularity of deep convolutional neural networks, researchers have mainly used deep, abstract, semantic information for FGVC, while shallow, detailed information has been neglected. This work proposes a cross-layer mutual attention learning network (CMAL-Net) to solve the above problems. Specifically, this work views the shallow to deep layers of CNNs as "experts" knowledgeable about different perspectives. We let each expert give a category prediction and an attention region indicating the found clues. Attention regions are treated as information carriers among experts, bringing three benefits: (*i*) helping the model focus on discriminative regions; (*ii*) providing more training data; (*iii*) allowing experts to learn from each other to improve the overall performance. CMAL-Net achieves state-of-the-art performance on three competitive datasets: FGVC-Aircraft, Stanford Cars, and Food-11. The source code is available at https://github.com/Dichao-Liu/CMAL

1. Introduction

Fine-grained visual classification (FGVC) aims to dependably distinguish visually similar categories, such as different models of airplanes [1] or cars [2]. FGVC has great potential value for many real-world applications. However, it is also extremely challenging, and the current classification accuracy still needs to be improved for large-scale practical usage. The existing FGVC approaches are mainly facing three problems: (*i*) The inter-class similarity and intra-class variation; (*ii*) Limited amount of training data. Collecting and labeling images for the FGVC task usually requires expert knowledge, and it is hard to create large-scale FGVC datasets; (*iii*) The neglect of the low-level information. Deep convolutional neural networks (CNNs) have become the dominant tool for handling the FGVC task. With the increase of the depth, CNNs focus more on high-level, abstract, and semantic information, while low-level and explicit information is neglected.

The first problem is the inherent characteristic of the FGVC task, and locating attention regions (i.e., the local regions containing discriminative clues) is generally considered to be the key to mitigating the negative effect caused by intra-class variation and inter-class similarity. However, attention regions are very hard to define and locate, and the attention information tends to vary from object to object. The second problem involves the difficulty of collecting and labeling the samples for FGVC. Data augmentation is a common strategy to solve the issue of lacking training data. Data augmentation refers to the approaches for increasing the amount of data by adding slightly changed copies of current data or creating new synthetic data from existing data. Most of the previous FGVC studies use random data augmentation to increase the training data, such as generating new samples by randomly cropping certain regions from the original images. However, random data augmentation possibly introduces some noises and contains certain redundancy information.

To the best of our knowledge, the third problem is seldom discussed in previous FGVC research. With the dominion of the CNNs in computer vision tasks, existing studies mainly develop FGVC algorithms based on the state-of-the-art CNN architectures, such as ResNet [5], Res2NeXt [6], TResNet [7], and etc. The shallow layers

☆☆ This work is supported by PhD Professional Toryumon Program of Nagoya University, Japan.

* Corresponding author.
E-mail addresses: dichao.liu@navier.co.jp, dichao_liu@outlook.jp (D. Liu), zlj@nagoya-u.jp (L. Zhao), ywang@fc.ritsumei.ac.jp (Y. Wang), jien@fc.ritsumei.ac.jp (J. Kato).
[1] Yu Wang is now with the Center for Information and Communication Technology, Hitotsubashi University, Tokyo 186-8601, Japan.

https://doi.org/10.1016/j.patcog.2023.109550
0031-3203/

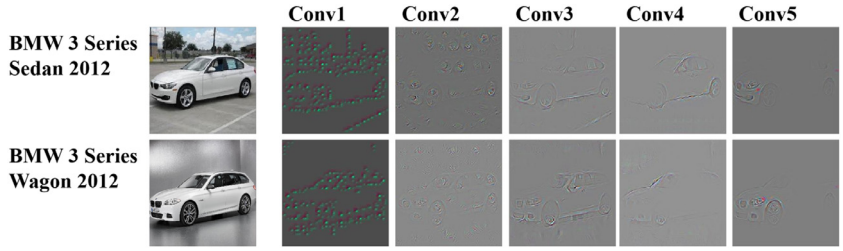


Fig. 1. Visualization results of guided backpropagation (GB) [3] implemented with an AlexNet [4] trained on the Stanford Cars dataset [2]. GB visualizes the part of an image that most activates the filters of a certain layer. Deep CNN layers abstract the low-level information learned by shallow layers and focus on semantically meaningful regions. However, deep CNN layers lose some details.

of CNNs learn basic patterns like varied orientations, curves, parallel lines, circles, and so on, while the deep layers encode such patterns to capture more abstract and semantically relevant information like automobile headlights, car logos, and so on. Existing state-of-the-art CNN designs often add classifiers on top of the deepest layers within the CNNs since deeper layers inherit the information learned by shallower layers through forward propagation and can provide better accuracy than shallower layers. However, we argue that the low-level detailed information learned by the layers shallower than the deepest layer can also provide effective clues for the FGVC task. For example, as shown in Fig. 1, The brand logo and headlights are generally important clues to identify the model of a car, and the deepest convolution layer focuses mainly on such discriminative and semantically meaningful information. However, some low-level but potentially useful details are lost in the deepest layer, such as the local contour orientations (e.g., the contour orientation of the roof at the back of the vehicle is an important cue to distinguish a sedan from a wagon). Such details learned by shallower layers are abstracted into semantic information by deeper layers, and therewith the details are largely lost.

In this paper, we propose a novel cross-layer mutual attention learning network (CMAL-Net) to address the three above-mentioned problems together. In this paper, we regard the layers of a CNN (e.g., ResNet50 [5], Res2NeXt50 [6], etc.) from shallow to deep as several "experts" who have knowledge from different perspectives. The shallow experts (i.e., the experts consisting of shallow layers) are knowledgeable about low-level detailed information. The deep experts (i.e., the experts covering into deep layers) are knowledgeable about high-level abstract semantic information. From shallow to deep, each expert learns with the prior knowledge from the previous expert. An ordinary convolutional neural network can be regarded as using only the deepest expert to make a prediction. In contrast, CMAL-Net lets each expert give a prediction of the categorical label and attention region. The attention region is predicted based on the attention maps generated inside each expert, which does not require manual region annotations or sophisticated attention-locating architecture.

CMAL-Net is trained in multiple steps at each iteration. At each step, we train one of the experts or the fusion of all the experts. For one expert, the attention regions proposed by the other experts are treated as possible data augmentations to choose from. Concretely, we start with the deepest expert, which inherits the prior knowledge of the previous ones and generally has better accuracy than the previous ones. Note that the training of the deepest experts also updates the parameters of the other experts. Thus, we can obtain the attention regions predicted by all the experts at the first step. Those attention regions are treated as augmentations to the original input for other steps. We gradually move on to shallow experts, and then, train the fusion of all experts.

CMAL-Net brings three benefits: (*i*) The attention regions proposed by each expert help to locate the discriminative clues and avoid inter-class similarity and intra-class variation; (*ii*) Treating the attention regions as data augmentations not only solves the problem of limited training data, but also avoids introducing unfavorable noise (e.g., discriminative parts are occluded); (*iii*) The attention regions are carriers of the experts' "specialized knowledge" since they show how the experts link certain regions of an image to the category prediction. By way of analogy, the attention regions are like notes made by the experts marking out what the experts consider to be the key clues. The "specialized knowledge" can be passed among the experts by using the attention region proposed by one expert as the input of another expert. Namely, in CMAL-Net, the experts learn from each other and enhance each other.

Our contributions : (*i*) We propose a novel framework that generates attention regions based on the information learned from varying depths of layers inside a single CNN backbone. The generated attention regions show the cues found by specific layers, and are used as a data augmentation to solve the lack of training data and allow deep and shallow layers to learn from each other to improve the overall classification accuracy. (*ii*) The proposed approach addresses three problems together: inter-class similarity and intra-class variation, lack of training data, and ignorance of low-level, detailed information. (*iii*) We achieve state-of-the-art performances on three standard and competitive FGVC datasets: FGVC-Aircraft [1], Stanford Cars [2], and Food-11 [8].

2. Related studies

2.1. Fine-grained image classification

In the FGVC field, attention learning has always been the dominant theme [9], which captures discriminative clues and can help the models combat the adverse effects caused by the inherent inter-class similarity/intra-class variation and understand the semantic importance of the local objects. For example, Zhang *et al.* [10] proposed the Sequentially Diversified Networks (SDNs) by constructing multiple lightweight sub-networks inserted into the backbone network to enable information interaction among local regions of the fine-grained image. SDNs jointly advanced the diversity of spatial attention, greatly contributing to effectively learning diverse representations. Niu *et al.* [11] investigated the attention-learning process from the view of human visual recognition mechanisms, in which the attention regions are temporally perceived via the attention-shift mechanism. They proposed the Attention-Shift based Deep Neural Network (AS-DNN) to find the attention regions and encode the semantic correlations among the found attention regions iteratively, which effectively boosts the classification performance. Du *et al.* [12] focused on which granularities of attention regions are most effective and effectively fuse the information across different granularities. They proposed a progressive training strategy to fuse multi-granularity attention features and a random jigsaw patch creator to force the model to realize attention information at specific granularities.

The above-mentioned studies have greatly advanced the development of image recognition research and are very inspiring for us. However, unlike the above studies, our work focuses on using attention regions predicted by layers of different depths to mark the cues they learned. These attention regions are used as both information carriers and data augmentation, allowing layers of different depths to learn from each other's knowledge to improve overall performance.

In addition to methods based on attention mechanisms, second-order pooling methods are also an important class of approaches in the field of FGVC. Second-order pooling approaches make use of the second-order statistics of deep features to compose powerful representations. Lin *et al.* [13] combined the feature maps respectively obtained by two CNNs at each location with the matrix outer product. Then the processed features are averaged to obtain a bilinear feature representation, effectively capturing subtle clues. Zheng *et al.* [14] proposed a deep bilinear transformation (DBT) block, which can be embedded into intermediate layers of CNNs and yield bilinear features inside CNNs. Wang *et al.* [15] utilized the covariance among deep features to construct representations and added matrix power normalization into the learning of global covariance pooling, which both improves training speed and decreases model complexity. These studies have shown that second-order pooling can capture rich statistics of deep features. However, the second-order pooling methods introduce many parameters compared to the traditional pooling method, increasing the computational cost and making optimization more difficult. In our experiments, second-order pooling does not provide more benefits than global max pooling for the designed network.

2.2. Shallow and deep features learned by neural networks

The Different Levels of Information Captured by the Deep and Shallow Layers. Convolutional neural networks (CNNs) have demonstrated outstanding performance on various image classification tasks. To understand why the CNNs perform so well, researchers have made many efforts to explore what CNNs have learned with their varying depths of layers. Zeiler *et al.* [16] proposed a multi-layered Deconvolutional Network (Deconvnet) to gain insight into the function of intermediate feature layers. They found that shallow layers learn low-level details while deep layers learn high-level semantic information. Jiang *et al.* [17] proposed the LayerCAM, which indicates the discriminative regions used by the different layers of a CNN to predict a specific category. The visualization results of their work clearly show that the CNN layers from shallow to deep gradually move the focus from local details to semantic regions. Our work is inspired by these previous studies to a certain extent.

Multiple Classifiers Based on Shallow and Deep Layers. To our knowledge, Lee *et al.* [18] first proposed developing multiple classifiers based on shallow and deep layers to improve performance. They designed the deeply-supervised nets (DSN) by introducing companion objectives to the intermediate layers. In this way, DSN improved the transparency and discernment of the intermediate layers to the classification task and helped solve the gradient vanishing problem. Inspired by [18], some recent researchers have applied the idea of adding auxiliary classifiers in the middle layers to their own studies. Çaylı *et al.* [19] added an auxiliary classifier to a long short-term memory (LSTM) model, which allows the gradient flow to reach the bottom layers during the recurrent learning process and improves the training performance of LSTM. Peng *et al.* [20] used the image-level classification task as a co-supervision task to improve the object detection task's performance, and following [18], they added the image-level supervision to mid-level features at different layers. The differences between our approach and the above methods can be summarized as: (*i*) The above methods only focus on adding classifiers in the middle layer to mitigate the gradient vanishing problem and improve the training process. Instead, we focus on the intercommunication and mutual learning of the captured cues between different layers to improve the overall performance. (*ii*) For the FGVC task of interest in this paper, we integrate attention learning into the training process of the multiple classifiers. (*iii*) As pointed out by Huang *et al.* [21], adding classifiers to shallow layers and compelling them to learn categorization violate neural networks' original feature extraction procedure, harming the overall performance. It is because shallow-layer-based classifiers require shallow layers to capture abstract semantic information, while deep-layer-based classifiers require shallow layers to capture concrete detail information. To solve this problem, we apply a multi-step training strategy to avoid competition between deep and shallow classifiers. The optimization of overall classification is set as the last step in each iteration to avoid the optimization of shallow layers dominating the learning process.

3. Approach

3.1. Expert construction

In this subsection, we introduce the details of how we construct the experts from shallow to deep. Let $\mathcal{B}$ be the backbone CNN, which can be any state-of-the-art CNNs, such as ResNet50 [5], Res2NeXt50 [6], etc. $\mathcal{B}$ has M layers, and $\{l_1, l_2, ..., l_m, ..., l_M\}$ denote the layers of $\mathcal{B}$ from shallow to deep (excluding the fully-connected classifier). $\{e_1, e_2, ..., e_n, ..., e_N\}$ are N experts based on the M layers. Each expert consists of the layers from the first layer l_1 to a certain layer of the M layers. For example, e_n consists of the layers from l_1 to l_{m_n}, and $1 \leqslant m_n \leqslant M$. The experts $\{e_1, e_2, ..., e_n, ..., e_N\}$ gradually cover deeper layers of $\mathcal{B}$, and the deepest expert e_N covers all the layers from l_1 to l_M.

Let $\{x_1, x_2, ..., x_n, ..., x_N\}$ denote the the intermediate feature maps generated by $\mathcal{B}$ for the experts $\{e_1, e_2, ..., e_n, ..., e_N\}$, respectively. $x_n \in \mathbb{R}^{H_n \times W_n \times C_n}$, and H_n, W_n and C_n denote the height, width and the number of channels, respectively. We use a set of functions $\{F_1(.), F_2(.), ..., F_n(.), ..., F_N(.)\}$ to respectively compress $\{x_1, x_2, ..., x_n, ..., x_N\}$ into 1D vectorial descriptors $\{v_1, v_2, ..., v_n, ..., v_N\}$, and $v_n \in \mathbb{R}^{C_v}$. C_v denotes the length of the 1D vectorial descriptors, and the 1D vectorial descriptors given by different experts have the same length. The $F_n(.)$ for processing x_n is defined as:

$$v_n = F_n(x_n) = f^{GMP}(x''_n), \tag{1}$$

$$x''_n = f^{Elu}(f^{bn}(f^{conv}_{3\times3\times\frac{C_v}{2}\times C_v}(x'_n))), \tag{2}$$

$$x'_n = f^{Elu}(f^{bn}(f^{conv}_{1\times1\times C_n\times\frac{C_v}{2}}(x_n))), \tag{3}$$

where $f^{GMP}(.)$ denotes the Global Max Pooling. $f^{conv}(.)$ illustrates the 2D convolution operation by its kernel size. For example, $f^{conv}_{3\times3\times C_n\times\frac{C_v}{2}}(.)$ means a 2D convolution operation whose kernel size is $3 \times 3 \times C_n \times \frac{C_v}{2}$ (3×3 is the spatial size, C_n is the number of input channels, and $\frac{C_v}{2}$ is the number of output channels). $f^{bn}(.)$ denotes batch normalization operation, and $f^{Elu}(.)$ denotes Elu operation. x'_n and x''_n are intermediate feature maps generated by e_n. Thereinto, $x''_n \in \mathbb{R}^{H_n \times W_n \times C_v}$ is used for generating the attention region of e_n (details in Subsection 3.2).

Then, let $\{p_1, p_2, ..., p_n, ..., p_N\}$ denote the prediction scores given by different experts, and we obtain the prediction scores as $p_n = f^{clf}_n(v_n)$, where $f^{clf}_n(.)$ denotes a fully connected layer-based classifier.

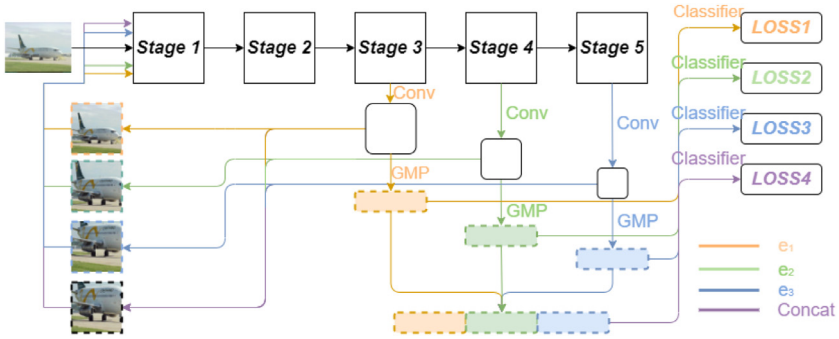


Fig. 2. Network structure. As an example for explanation, this figure illustrates a CMAL-Net constructed by adding three experts e_1, e_2, e_3, on a 5-stage backbone CNN (e.g., ResNet50). The workflow of each expert and the concatenation of the experts are shown in different colors. Each expert takes as input the feature maps from a particular layer and outputs a categorical prediction together with an attention region that is used as the data augmentation of other experts. This architecture is trained with multiple steps for each iteration. In the first step, we train the deepest expert e_3 and then go into shallower layers. Thereafter, in the last step, we train the concatenation of the experts to improve the overall performance.

Besides the prediction scores given by the individual experts, we also generate an overall prediction score by aggregating the information learned by different experts. Specifically, we first concatenate $\{v_1, v_2, ..., v_n, ..., v_N\}$ to be an overall descriptor v_{oval} as: $v_{oval} = f^{concat}(v_1, v_2, ..., v_n, ..., v_N)$, where $f^{concat}(.)$ denotes concatenation operation. Then v_{oval} is processed into an overall prediction score p_{oval} by a fully connected layer-based classifier as $p_{oval} = f^{clf}_{oval}(v_{oval})$.

Modern CNN architectures are generally composed of stages [5,7], which refers to groups of layers operating on the feature maps of the same spatial size. The spatial size of the feature maps decreases from the shallow to deep stages. For example, the layers of ResNet50 (excluding the fully-connected classifier) are grouped into 5 stages. Given an input image with spatial size 448×448 to ResNet50, the spatial sizes of the output feature maps of the layers belonging to the five stages are 224×224, 112×112, 56×56, 28×28, 14×14, respectively from shallow to deep. As shown in Fig. 2, we use stage as the unit of constructing the experts.

3.2. Attention region prediction

As defined above, x''_n denotes an intermediate feature map generated by the experts e_n, and $x''_n \in \mathbb{R}^{H_n \times W_n \times C_v}$. Let us assume that the classification problem is K-class, and $k_n \in \{1, 2, ..., K\}$ is the category predicted by e_n. The generation of the attention region proposed by e_n starts by producing the class activation map (CAM) [22] for the category k_n based on x''_n. A CAM for a particular class specifies the discriminative image areas used by the CNN to recognize that class. The CAM Ω_n ($\Omega_n \in \mathbb{R}^{H_n \times W_n}$) produced by the expert e_n is defined as:

$$\Omega_n(\alpha, \beta) = \sum_{c=1}^{C_v} p^c_n x''^c_n(\alpha, \beta), \tag{4}$$

where the coordinates (α, β) denote the spatial location of x''_n and Ω_n. p_n denotes the parameters of $f^{clf}_n(.)$ corresponding to the predicted category k_n. $\Omega_n(\alpha, \beta)$ directly indicates the importance of the activation at spatial location (α, β) leading to the "decision" of the expert e_n on categorizing an image to category k_n. Each unit on the intermediate feature map x''_n is activated by certain visual patterns within its receptive field. The CAM is essentially a weighted linear sum of these visual patterns' occurrence at different spatial locations [22]. By upsampling the CAM to the input image's size, we can be informed of the areas of the image that appear to be most relevant to the category k_n from the expert e_n's perspective of view. Thus, after obtaining Ω, we generate an attention map $\tilde{\Omega}_n \in \mathbb{R}^{H_{in} \times W_{in}}$ (H_{in}, W_{in} are the height and width of the input image, respectively) by upsampling Ω_n using a bilinear sampling kernel. Thereafter, $\tilde{\Omega}_n$ is applied with min-max normalization, and each spatial element of the normalized attention map $\tilde{\Omega}^{norm}_n$ is given by

$$\tilde{\Omega}^{norm}_n(\alpha, \beta) = \frac{\tilde{\Omega}_n(\alpha, \beta) - \min(\tilde{\Omega}_n)}{\max(\tilde{\Omega}_n) - \min(\tilde{\Omega}_n)}. \tag{5}$$

We can find and crop the regions that the expert e_n considers discriminative by utilizing the normalized attention map $\tilde{\Omega}^{norm}_n$ as guidance. Concretely, we first generate a mask $\tilde{\Omega}^{mask}_n$ by setting the elements in $\tilde{\Omega}^{norm}_n$ to 1 for the values greater than a threshold t ($t \in [0, 1]$) and 0 for the others. Namely, each spatial element of the mask $\tilde{\Omega}^{mask}_n$ is given by

$$\tilde{\Omega}^{mask}_n(\alpha, \beta) = \begin{cases} 1, & if\ \tilde{\Omega}^{norm}_n(\alpha, \beta) - t > 0 \\ 0, & if\ \tilde{\Omega}^{norm}_n(\alpha, \beta) - t <= 0. \end{cases} \tag{6}$$

We locate a bounding box that can cover all the positive regions of $\tilde{\Omega}^{mask}_n$ and crop this region from the input image. Then we upsample the cropped region to the input image's size and treat the upsampled attention region $\mathcal{A}_n$ as the attention region predicted by e_n as well as data augmentation for other experts.

Besides the attention regions proposed by individual experts, we also generate an overall attention region $\mathcal{A}_{oval}$ by summing up the attention information learned by different experts. Specifically, the generation of $\mathcal{A}_{oval}$ starts by averaging the normalized attention maps of different experts to be the overall attention map $\tilde{\Omega}_{oval}$. Each spatial element of the overall attention map $\tilde{\Omega}_{oval}$ is given by

$$\tilde{\Omega}_{oval}(\alpha, \beta) = \frac{1}{N} \sum_{n=1}^{N} \tilde{\Omega}_n(\alpha, \beta). \tag{7}$$

Thereafter, similar to the process of generating the attention regions with individual experts, $\tilde{\Omega}_{oval}$ is applied with min-max normalization and the result is denoted as $\tilde{\Omega}^{norm}_{oval}$. Then the elements of $\tilde{\Omega}^{norm}_{oval}$ are set to 1 or 0 based on the same threshold t. Lastly, the region covering all the positive values is located, and the same region of the input image is cropped and unsampled to be the overall attention region $\mathcal{A}_{oval}$, which has the same size as the input image.

3.3. Multi-step mutual learning

We train the experts in a progressive multi-step strategy with cross-entropy loss. In the early steps, we train these experts one by one, which allows them to "concentrate on" learning the clues of their own domain without being distracted by other experts. In

the last two steps, the experts work together to learn effective information from the attention regions and the raw image, respectively. Specifically, as shown in Algorithm 1, each iteration of the

Algorithm 1 Multi-step mutual learning

Require: Given a dataset $\mathcal{D} = \{(\text{input}^i, \text{target}^i)\}_{i=1}^{I}$ (I is the total number of batches in $\mathcal{D}$), and N experts $\{e_1, e_2, ..., e_n, ..., e_N\}$. $\mathcal{L}_{cls}(.)$ denotes the cross entropy loss for the classification task.

for epoch = 1 to number_of_epochs **do**
  for (input, target) in $\mathcal{D}$ **do**

    ▷ Train the experts from deep to shallow with data augmentation by multiple steps.
    $x_N, \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n, ..., \mathcal{A}_N, \mathcal{A}_{oval}\} \leftarrow e_N(\text{input})$
    $v_N \leftarrow F_N(x_N)$
    $p_N \leftarrow f_N^{clf}(v_N)$
    $\mathcal{L}_N \leftarrow \mathcal{L}_{cls}(p_N, \text{target})$
    BACKPROP$(\mathcal{L}_N)$
    for n = $N-1$ downto 1 by -1 **do**
      $\text{input}_n \leftarrow$ Randomly_choose_from
      $(\{\text{input}, \mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_{n-1}, \mathcal{A}_{n+1}, ..., \mathcal{A}_N\})$
      $x_n \leftarrow e_n(\text{input}_n)$
      $v_n \leftarrow F_n(x_n)$
      $p_n \leftarrow f_n^{clf}(v_n)$
      $\mathcal{L}_n \leftarrow \mathcal{L}_{cls}(p_n, \text{target})$
      BACKPROP$(\mathcal{L}_n)$
    end for

    ▷ Train the experts and their concatenation with $\mathcal{A}_{oval}$ in one go.
    for n = 1 to N **do**
      $x_n^{\mathcal{A}} \leftarrow e_n(\mathcal{A}_{oval})$
      $v_n^{\mathcal{A}} \leftarrow F_n(x_n^{\mathcal{A}})$
      $p_n^{\mathcal{A}} \leftarrow f_n^{clf}(v_n^{\mathcal{A}})$
    end for
    $v_{oval}^{\mathcal{A}} = f^{concat}(v_1^{\mathcal{A}}, v_2^{\mathcal{A}}, ..., v_N^{\mathcal{A}})$
    $p_{oval}^{\mathcal{A}} \leftarrow f_{oval}^{clf}(v_{oval}^{\mathcal{A}})$
    $\mathcal{L}^{\mathcal{A}} \leftarrow \mathcal{L}_{cls}(p_1^{\mathcal{A}}, \text{target}) + \mathcal{L}_{cls}(p_2^{\mathcal{A}}, \text{target}) + ... + \mathcal{L}_{cls}(p_N^{\mathcal{A}}, \text{target}) + \mathcal{L}_{cls}(p_{oval}^{\mathcal{A}}, \text{target})$
    BACKPROP$(\mathcal{L}^{\mathcal{A}})$

    ▷ Train the concatenation of the experts with the raw input.
    for n = 1 to N **do**
      $x_n \leftarrow e_n(\text{input})$
      $v_n \leftarrow F_n(x_n)$
    end for
    $v_{oval} \leftarrow f^{concat}(v_1, v_2, ..., v_N)$
    $p_{oval} \leftarrow f_{oval}^{clf}(v_{oval})$
    $\mathcal{L}_{oval} \leftarrow \mathcal{L}_{cls}(p_{oval}, \text{target})$
    BACKPROP$(\mathcal{L}_{oval})$
  end for
end for

training contains $N+2$ steps, and in the first N steps, we gradually train each expert from deep to shallow. In the first step, we train the deepest expert e_N. Since the training of N involves the experts shallower than e_N, we are also able to generate the attention regions proposed by all the experts and the overall attention region $\{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n, ..., \mathcal{A}_N, \mathcal{A}_{oval}\}$ at this step. These attention regions carry the "specialized knowledge" of the experts by marking the basis on which each expert made its classification judgment.

Then from step 2 to step N, we gradually move on to shallow experts with the proposed data augmentation strategy, which we refer to as mutual data augmentation (MDA). When training an expert, MDA randomly selects one input from an image pool consisting of the raw input and the attention regions proposed by experts other than this expert. Deep experts conclude the information learned by shallow experts hierarchically and generally enable better descriptions of latent concepts than shallow experts for the classification task. The attention regions proposed by deep experts help shallow experts learn the semantic visual cues (e.g., headlights) found by deep experts in the input image. On the other hand, though the low-level information learned by shallow experts has been transmitted to deep experts via forward propagation, many details are lost because low-level information is abstracted. The attention regions proposed by shallow experts help deep experts learn the low-level visual cues (e.g., local edge orientations, object textures, etc.) found by deep experts in the input image. Experts are trained step by step from deep towards shallow, forcing the shallower experts to make judgments based on the clues learned by deeper experts, rather than simply acting as information providers for the deeper experts.

At step $N+1$, we train all the experts and their concatenation with the overall attention region $\mathcal{A}_{oval}$ in one go. $\mathcal{A}_{oval}$ is proposed by all experts together and should contain the clues considered important by the collective of experts. This step brings all experts together to amplify and study the attention information they have jointly acquired for extracting more fine-grained features. At step $N+2$, we train the concatenation of all the experts with the raw input to ensure the parameters of $f_{oval}^{clf}(.)$ fit the resolution of the objects in the original input. Note that as discussed in Subsection 2.2, the tasks of shallow and deep layers compete to some extent, and thus we optimize the tasks step by step. Each step of optimization has different and to-some-extent competing tasks, different inputs, and trains different parts of the network. Thus, completing all the tasks described in Algorithm 1 is a complete iteration.

As shown in Fig. 2, CMAL-Net has $N+1$ classifiers. Namely, given an input image at the inference phase, the proposed architecture can produce $N+1$ prediction scores. In the implementation of inference, for each image, we successively feed the raw input and overall attention region into CMAL-Net to obtain a total of $2 \times (N+1)$ prediction scores. The final prediction score for the inference is calculated by averaging the $2 \times (N+1)$ prediction scores. This inference strategy maximizes the classification accuracy of the trained model because of two facts: (*i*) the prediction scores given by each expert and the overall prediction score can provide complementary information; (*ii*) the information learned from the raw input and overall attention region can provide complementary information.

4. Experiments

4.1. Datasets and implementation details

Datasets. The experiments in this section involve three standard and very competitive benchmarks, namely FGVC-Aircraft [1], Stanford Cars [2], and Food-11 [8]. As shown in Table 1, FGVC-Aircraft [1] is an aircraft image dataset spanning 100 categories of aircraft models. There are 6,667 training images and 3,333 testing images in FGVC-Aircraft. Stanford Cars dataset [2] is an image dataset with photos of 196 car models, consisting of 8,144 training images and 8,041 testing images. Food-11 [8] contains the food images grouped in 11 meal categories. This dataset has 9,866 training images and 3,347 testing images. Note that although bounding boxes or part annotations are available with these datasets, the proposed method does not use this extra supervision information

Table 1
Details of the Datasets Used in This Paper.

	Dataset Content	Categories	Training Images	Testing Images
FGVC-Aircraft	Aircraft Models	100	6,667	3,333
Stanford Cars	Car Models	196	8,144	8,041
Food-11	Dishes	11	9,866	3,347

Table 2
Ablation studies on the different pooling methods, contribution of MDA, and the inference strategy.

	Test Input	p_1	p_2	p_3	p_{oval}	Avg. of Predictions	Fusion of Raw + $\mathcal{A}_{oval}$/Aug.*
Train with	Raw	90.5%	93.9%	92.7%	94.2%	94.4%	94.7%
MDA	$\mathcal{A}_{oval}$	90.3%	93.9%	92.5%	94.1%	94.3%	
Train with	Raw	87.0%	92.5%	92.2%	92.6%	93.0%	93.5%
MDA (GAP)	$\mathcal{A}_{oval}$	87.1%	92.5%	91.9%	92.4%	93.1%	
Train with	Raw	88.9%	92.5%	91.4%	92.8%	93.2%	93.2%
MDA (BP)	$\mathcal{A}_{oval}$	88.8%	92.6%	91.4%	92.9%	93.2%	
Train with	Raw	88.6%	93.1%	91.6%	93.0%	93.3%	93.5%
MDA (GCP)	$\mathcal{A}_{oval}$	88.8%	93.0%	91.7%	93.2%	93.4%	
Baseline 1	Raw	88.7%	92.5%	91.2%	92.6%	92.8%	-
Baseline 2	Raw	79.6%	88.7%	89.7%	90.6%	90.6%	91.8%
	Aug.	79.8%	86.6%	84.5%	86.8%	90.5%	
Baseline 3	Raw	89.4%	92.8%	91.8%	92.9%	92.8%	93.1%
	Aug.	89.6%	92.9%	91.9%	92.9%	93.0%	
Baseline 4	Raw	89.6%	92.9%	91.9%	92.9%	92.9%	93.0%
	Aug.	89.9%	92.8%	92.1%	93.0%	92.8%	
Baseline 5	Raw	89.5%	92.1%	90.8%	92.5%	92.6%	-
Baseline 6	Raw	89.1%	92.3%	91.8%	93.1%	93.6%	-
Baseline 7	Raw	89.4%	92.7%	91.7%	93.3%	93.6%	-
Baseline 8	Raw	71.7%	88.6%	91.2%	92.3%	92.3%	-
Baseline 9	Raw	89.8%	92.7%	92.4%	93.6%	93.7%	-

* Aug. represents the image generated after the raw input has been applied with the corresponding data augmentation. ** Unless otherwise indicated, the classifier is built by default on top of the global maximum pooling (GMP) of features.

for the training. The bounding boxes are only used for analyzing the attention-capturing ability (Subsection 4.5).

Details of the experts. To verify the classification performance as well as the generality of the proposed approach, we evaluate it with three different backbone CNNs: ResNet50 [5], Res2NeXt50 [6], and TResNet-L [7]. These backbone CNNs all have five stages, on which we build three experts: e_1 covers the layers from stage 1 to stage 3; e_2 covers the layers from stage 1 to stage 4; e_3 covers the layers from stage 1 to stage 5.

Training details. For the experiments in this work, we train the models using Stochastic Gradient Descent (SGD) with the epoch number of 200, momentum of 0.9, weight decay of 5×10^{-4}, and a mini-batch size of 16. Unless otherwise indicated, the learning rate is set as 0.002 with cosine annealing [23]. We set the input size as 448×448 for ResNet50 and Res2NeXt50, following the common settings in previous studies [24,25]. The input size of TResNet-L is set as 368×368 following Ridnik *et al.* [7], who are the authors of this architecture. The threshold t is set to 0.5 (except in the ablation study over it). Note that the previous state-of-the-art accuracies listed in Tables 7 and 8 (marked by underlining) are cited from their original papers.

4.2. Training and testing statistics during the learning process

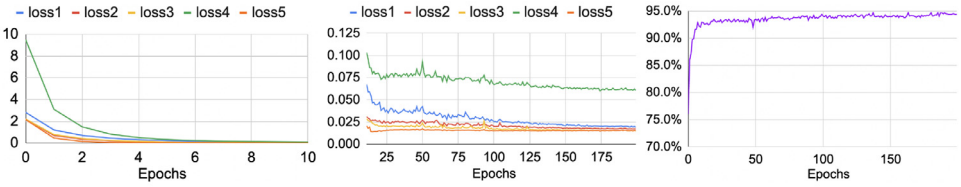
This subsection evaluates the training and testing statistics during the learning process. Following a common evaluation strategy, we assess the training loss and test accuracy for each epoch, and the results are presented in Fig. 3. The experiment in this subsection uses a CMAL-Net with ResNet50 as the backbone network and FGVC-Aircraft as the dataset. As shown in Fig. 3, the training loss decreases sharply in the first ten epochs and then drops to saturation slowly and gradually. Analogously, the testing accuracy rises rapidly in the early epochs and then slowly and gradually until it reaches saturation in the late epochs. These results show that the training of the proposed CMAL-Net is stable, and the various losses do not pull on each other to the extent that the training or test statistics fluctuate abnormally.

4.3. Ablation studies

To comprehensively analyze our method, we implement ablation studies of the design choices. This subsection chooses the FGVC-Aircraft dataset for experiments and ResNet50 as the backbone network.

Evaluation of Different Pooling Methods for Building Experts. As introduced in Section 3, each expert uses a global max pooling (GMP) layer to aggregate intermediate feature maps into a descriptor. Here we evaluate the effectiveness of this design. We use global average pooling (GAP), bilinear pooling (BP) [13], and global covariance pooling (GCP) [15] to replace GMP, respectively, and observe the change in accuracy. As shown in Table 2, the accuracy obtained with GMP is 1.2%–1.5% higher than that obtained with the other pooling methods. We assume it is because GMP captures sharp features and behaves invariant to translation, scale, etc., which helps the model to capture attentional information from extensive data augmentation. To be specific, in CMAL-Net, different experts propose different attention regions where key objects may lie at different regional locations, and GMP can cope well with this situation. In addition, GMP can effectively reduce the parameters, which helps to prevent our model from falling into local optimal solutions in complex learning tasks due to difficulties in parameter optimization.

The contribution of mutual data augmentation (MDA). MDA randomly chooses inputs from an image pool consisting of the raw input images and attention regions. This strategy serves both as a way to pass on clues learned by different experts and as data augmentation. To verify the contribution of MDA, we compare the proposed CMAL-Net with nine baselines, all of which have the ex-



(a) Training loss (Part 1) (b) Training loss (Part 2) (c) Testing accuracy.

Fig. 3. Training and testing curves. The training curves are split into two parts for presentation, i.e., (a) epochs 1–10 and (b) epochs 11–200. This is because, in (a), the training loss decreases sharply from very high values, while in (b), the training loss decreases slowly until saturation, which makes it difficult to observe the respective characteristics of the two parts of the curves at the same scale. Losses 1–3 denote the loss values given by p_1–p_3 during the training, respectively. Loss 4 denotes the loss value given by p_{oval} on $\mathcal{A}_{oval}$. Loss 5 denotes the loss value given by p_{oval} on the raw input image. (c) shows the inference accuracy of different epochs.

act same architecture as CMAL-Net but are fed with different inputs during the multi-step training. During the training process, the number of steps, the order of the steps, and the part of the model trained in each step are set to be the same as CMAL-Net.

Baseline 1 is only fed with raw input images at all the steps. For Baseline 2, we apply random crops with a crop scale of 0.5 on the raw input and resize the cropped region to 448×448. Then we use the resized crops to replace the attention regions during the training. For Baseline 3, we first pad the raw input with a padding size of 16 and then random crop a 448×448 region. Then we use the crops to replace the attention regions during the training. For Baseline 4, We add random flipping to the data augmentation used in Baseline 3.

For Baseline 5–9, we apply the state-of-the-art data augmentation methods Mixup [26], CutMix [27], SaliencyMix [28], and CoMixup [29] and their random combination (one of the four data augmentation methods is randomly selected at each step) on the experts, respectively. Note that these state-of-the-art data augmentation approaches focus on convex combinations of data pairs instead of generating new data samples. Thus they cannot be used to fuse prediction scores derived from data augmentation with those derived from the raw image to improve accuracy further.

As shown in Table 2, whether the raw input or overall attention region is used as test input, the network trained with MDA is 0.9%–3.4% higher than Baselines 1–4 for different single prediction scores. Our method's best accuracy is 1.6%–2.9% higher than the best accuracy that Baselines 1–4 can achieve. Besides, a plain ResNet50 achieves 92.3% accuracy in our experiment (details in Table 4), which is lower than the accuracy achieved by the p_{oval} (92.6%) of Baseline 1. These results show that the multi-step training itself can slightly improve classification performance, and the MDA strategy has a huge contribution to improving classification accuracy. From the results of this experiment, we conclude that having the experts learn from each other about the cues they find can substantially improve classification accuracy. The best accuracy obtained with MDA is 1.0%–2.4% higher than that of Baselines 5–9, which use state-of-the-art data augmentation methods [26–29]. With only raw image as input, the accuracy obtained with MDA is 0.7%–2.1% higher than that of Baselines 5–9. MDA not only plays the role of data augmentation but also delivers attentional information between the different layers, which effectively improves accuracy.

The contribution of the inference strategy. As introduced above, for the testing procedure, the final prediction score is obtained by fusing the prediction scores given by different experts based on different types of inputs (i.e., raw input and the overall attention region $\mathcal{A}_{oval}$). It is necessary to verify the effectiveness of this inference strategy. As shown in Table 2, when the raw image is the input, the average of the prediction scores achieves 0.2%–3.9% higher accuracy than every single one of them. Similarly, when $\mathcal{A}_{oval}$ is the input, the average of the prediction scores achieves 0.2%–4.0% higher accuracy than every single one of them. Furthermore, the fusion of the prediction scores obtained with the raw image and $\mathcal{A}_{oval}$ is 0.3%–0.4% higher than each. These results verify the complementarity of the information learned by different experts based on different types of inputs. In the rest part of this paper, we report the accuracy obtained by fusing all the prediction scores. Moreover, note that the same inference strategy can also improve the accuracy of Baselines 2-5. However, as mentioned above, our best accuracy largely exceeds the maximum accuracy that baselines can achieve.

Table 3
The classification accuracy under different threshold (t) values.

t	0.1	0.3	0.5	0.7	0.9
Accuracy	93.2%	93.9%	**94.7%**	94.3%	93.2%

Table 4
Comparison with original networks on classification accuracy.

		FGVC-Aircraft	Stanford Cars
ResNet50	Original	92.3%	93.1%
	CMAL-Net	94.7%	94.9%
Res2NeXt50	Original	92.2%	93.2%
	CMAL-Net	94.2%	94.4%
TResNet-L	Original	90.4%	96.0%
	CMAL-Net	93.1%	97.1%

Impact of the threshold t. The threshold t affects the generation of the attention regions, and Table 3 shows the classification accuracy under different values of the threshold t. Setting t as 0.5 brings the highest accuracy. In the rest of this paper, we set $t = 0.5$ by default.

4.4. Comparison with original convolutional neural networks

This subsection compares the classification accuracy between the proposed approach and the original backbone CNNs to show that the proposed approach improves classification accuracy steadily on various backbone CNNs. This subsection chooses the FGVC-Aircraft dataset and Stanford Cars dataset for experiments and ResNet50, Res2NeXt50, and TResNet-L as the backbone networks. The results are shown in Table 4. On the FGVC-Aircraft dataset, the proposed approach improves 2.0%–2.7% over the three original CNNs. On the Stanford Cars dataset, the proposed approach improves 1.1%–1.8% over the three original CNNs. These results verify the effectiveness and generality of the proposed approach. Especially, the original TResNet-L achieves 96.0% accuracy on the Stanford Cars dataset, which is a very high accuracy. It is interesting to see there is still room for more than 1% improvement by our proposed method. This fact shows that even for a well-designed network architecture that achieves very high accuracy, the designed "mutual learning" between layers of different depths

Fig. 4. Visualization results. Each set of 6 images, from left to right, are the input image and visualization results based on $\tilde{\Omega}_{ori}^{norm}$, $\tilde{\Omega}_{1}^{norm}$, $\tilde{\Omega}_{2}^{norm}$, $\tilde{\Omega}_{3}^{norm}$, and $\tilde{\Omega}_{oval}^{norm}$.

Table 5
Attention capturing error of original ResNet50 and ResNet50-based CMAL-Net.

	Attention Capturing Error				
	$\tilde{\Omega}_{ori}^{norm}$	$\tilde{\Omega}_{1}^{norm}$	$\tilde{\Omega}_{2}^{norm}$	$\tilde{\Omega}_{3}^{norm}$	$\tilde{\Omega}_{oval}^{norm}$
FGVC-Aircraft	42.4%	28.0%	7.8%	7.6%	6.3%
Stanford Cars	65.7%	22.5%	6.1%	7.2%	5.9%

Table 6
Attention capturing recall of original ResNet50 and ResNet50-based CMAL-Net.

	Attention Capturing Recall				
	$\tilde{\Omega}_{ori}^{norm}$	$\tilde{\Omega}_{1}^{norm}$	$\tilde{\Omega}_{2}^{norm}$	$\tilde{\Omega}_{3}^{norm}$	$\tilde{\Omega}_{oval}^{norm}$
FGVC-Aircraft	57.6%	72.0%	92.2%	92.6%	93.6%
Stanford Cars	33.9%	77.4%	93.8%	92.7%	94.0%

can further improve the accuracy, which supports the theory on which our study is based.

4.5. Analysis on attention capturing

This subsection analyzes how the proposed CMAL-Net improves the attention-capturing ability over original CNNs. We use ResNet50 as the backbone and implement experiments on the FGVC-Aircraft and Stanford Cars.

First, we visualize the attention learned by the ResNet50-based CMAL-Net and original ResNet50 based on CAM [22]. For CMAL-Net, we can generate 4 heatmaps for each image, i.e., $\tilde{\Omega}_{1}^{norm}$, $\tilde{\Omega}_{2}^{norm}$, $\tilde{\Omega}_{3}^{norm}$, and $\tilde{\Omega}_{oval}^{norm}$. For the original ResNet50, since it has only 1 classifier for prediction, we generate 1 heatmap $\tilde{\Omega}_{ori}^{norm}$ based on the feature maps of the last convolutional layer. The results are shown in Fig. 4. $\tilde{\Omega}_{3}^{norm}$ and $\tilde{\Omega}_{ori}^{norm}$ are both generated based on the feature maps of the last convolutional layer inside the ResNet50 backbone, but $\tilde{\Omega}_{3}^{norm}$ captures much more comprehensive information than $\tilde{\Omega}_{ori}^{norm}$. For example, when recognizing aircraft models, $\tilde{\Omega}_{ori}^{norm}$ tends to only focus on a certain region in the aircraft body, while $\tilde{\Omega}_{3}^{norm}$ also captures other discriminative regions such as the tail unit. Besides, as the aggregation of $\tilde{\Omega}_{1}^{norm}$, $\tilde{\Omega}_{2}^{norm}$, and $\tilde{\Omega}_{3}^{norm}$, $\tilde{\Omega}_{oval}^{norm}$ also captures comprehensive attention.

Second, we quantitatively analyze the attention-capturing ability of CMAL-Net with attention-capturing error (ACE) and attention-capturing recall (ACR). The calculation of the ACE and ACR starts with generating an attention mask following the process defined as Eqs. (4), (5), and (6) in Section 3. The attention mask identifies the positive and negative pixels of the input image. Then, ACE is obtained by calculating the failure percentage of the images whose positive pixels have less than 50% attention precision [30] with the ground truth bounding box. Attention precision [30] is defined as the percentage of positive pixels that fall inside the ground truth bounding box among all the positive pixels. To obtain ACR, we first calculate the percentage of images in each category that have both the correct predicted label and an attention precision of no less than 50% with respect to the ground truth bounding box. Then ACR is obtained by averaging the evaluation results of all the categories. We compute the ACE and ACR obtained with $\tilde{\Omega}_{n}^{norm}$, $\tilde{\Omega}_{n}^{norm}$, $\tilde{\Omega}_{n}^{norm}$, $\tilde{\Omega}_{n}^{norm}$, and $\tilde{\Omega}_{n}^{norm}$ respectively. The results are shown in Tables 5 and 6. The ACE obtained with the heatmaps generated by ResNet50-based CMAL-Net is distinctively lower on both datasets than those obtained with the heatmaps generated by the original ResNet50 while the ACR is distinctively higher. The lower ACE indicates that the proposed approach enables the backbone to find important objects from the image rather than being distracted by background objects. The higher ACR indicates that the proposed approach enables the backbone to find cues that are correct and valid information for the classification task.

4.6. Illustration of the attention regions

Some examples of the attention regions proposed by different experts (i.e., $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$) and the overall attention region (i.e., $\mathcal{A}_{oval}$) are shown in Fig. 5. $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ are proposed by e_1, e_2, and e_3, respectively, based on their respective perspectives of views on what cues are important for classification. $\mathcal{A}_{oval}$ is proposed by the aggregation of $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$.

As can be observed from Fig. 5, from $\mathcal{A}_1$ to $\mathcal{A}_2$ to $\mathcal{A}_3$, important visual clues for classification are gradually zoomed in. It is because shallow layers are sensitive to low-level detail information scattered in various parts of the target object, while deep layers are sensitive to high-level semantic information concentrated in a particular part. As the attention regions are the carriers of the attention information learned by the experts, we can see how different experts have learned different perspectives of clues. $\mathcal{A}_{oval}$ reflects the overall attention information based on the attention information given by each expert and contains more comprehensive visual clues.

4.7. Comparison with State-of-the-Art approaches

This subsection compares our approach with very recent state-of-the-art approaches on three datasets: FGVC-Aircraft, Stanford Cars, and Food-11. The comparison results on FGVC-Aircraft and Stanford Cars are shown in Table 7. The comparison results on Food-11 are shown in Table 8. The accuracies obtained with all three backbone CNNs are listed in the tables.

FGVC-Aircraft: The proposed approach surpasses all the recent state-of-the-art methods and achieves the best performance with an accuracy of 94.7% with the ResNet50 backbone.

Stanford Cars: Using ResNet50 as the backbone, we achieve an accuracy of 94.9%, which is comparative to recent state-of-the-art approaches. When using TResNet-L as the backbone, our approach achieves an accuracy of 97.1% and surpasses all the recent state-of-the-art approaches with clear margins.

Table 7
Comparison with state-of-the-art methods on FGVC-Aircraft and Stanford Cars.

Approach	Backbone	FGVC- Aircraft	Stanford Cars
DBTNet-101 (NeurIPS, 2019 [14])	ResNet101	91.6%	94.5%
ImageNet + iNat on WS-DAN (ISVC, 2020 [31])	Inception V3	91.5%	-
SEF (Signal Processing Letters, 2020 [25])	ResNet50	92.1%	94.0%
MC Loss (TIP, 2020 [32])	B-CNN	92.9%	94.4%
GCL (AAAI, 2020 [33])	ResNet50	93.2%	94.0%
CIN (AAAI, 2020 [24])	ResNet101	93.3%	94.5%
DF-GMM (CVPR, 2020 [34])	ResNet50	93.8%	94.8%
LIO (CVPR, 2020 [35])	ResNet50	92.7%	94.5%
PMG (ECCV, 2020 [12])	ResNet50	92.8%	95.1%
iSQRT-COV-Net (TPAMI, 2021 [15])	ResNet101	91.4%	93.3%
B-CNN (TPAMI, 2021 [13])	VGG-M + VGG-D	84.1%	90.6%
Grafit (ICCV, 2021 [36])	EfficientNet-B7	-	94.7%
DeiT (ICML, 2021 [37])	DeiT-B	-	93.3%
NAT (TPAMI, 2021 [38])	NAT-M4	90.8%	92.9%
AutoFormer (ICCV, 2021 [39])	AutoFormer-S	-	93.4%
AS-DNN (Pattern Recognition, 2021 [11])	AS-DNN$_f$	92.3%	94.1%
MaskCOV (Pattern Recognition, 2021 [40])	ResNet50	-	94.0%
TransFG (AAAI, 2022 [9])	ViT-B_16	-	94.8%
ADCNN (Pattern Recognition, 2022 [41])	W-ResNet101	92.5%	91.3%
SDNs (Pattern Recognition, 2022 [10])	ResNet101	92.7%	94.6%
Ours	ResNet50	94.7%	94.9%
Ours	Res2Next50	94.2%	94.4%
Ours	TResNet-L	93.1%	97.1%

* The underline marks the accuracy reported in its original paper.

Table 8
Comparison with the state-of-the-art approaches on Food-11.

Approach	Backbone	Accuracy
Inception V3+Transfer learning (ICIIBMS, 2018 [42])	Inception V3	92.9%
ANN (Computers in Biology and Medicine, 2018 [43])	ResNet152	91.3%
Food-effective DCNN (JIT, 2018 [44])	Alexnet	86.9%
ResNet50+Transfer learning (DICTA, 2018 [45])	ResNet50	88.1%
Feature Fusion (IDAP, 2019 [46])	AlexNet+VGG16	89.3%
IOWA (Journal of Electronic Imaging, 2019 [47])	AlexNet+ResNet50+ VGG19+ GoogLeNet	90.6%
LNAS (Mathematics, 2021 [48])	LNAS-net	89.1%
Ours	ResNet50	96.3%
Ours	Res2Next50	96.5%
Ours	TResNet-L	95.4%

* The underline marks the accuracy reported in its original paper.

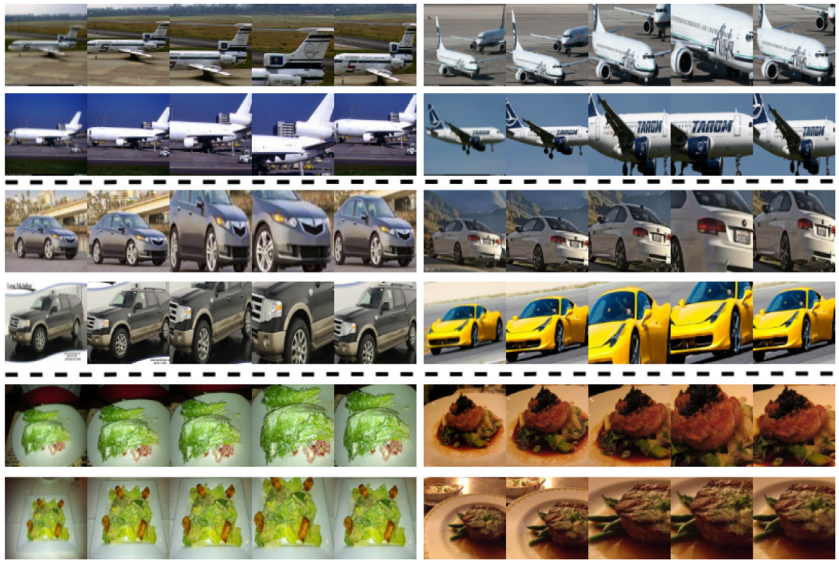


Fig. 5. Examples of attention regions. Each set of 5 images, from left to right, shows the raw input image, $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_{ovall}$, respectively.

Table 9
Comparison of inference cost.

	Input Size	Parameters	MACs	Inference Time
SEF [25]	448×448	24M	16.5G	4.378ms
MC Loss [32]	448×448	24M	16.5G	4.060ms
LIO [35]	448×448	24M	16.5G	4.174ms
PMG [12]	448×448	45M	37.4G	7.720ms
iSQRT-COV-Net [15]	448×448	46M	40.4G	9.487ms
DeiT [37]	384×384	86M	49.4G	9.652ms
NAT [38]	252×252	5M	0.4G	4.843ms
Autoformer [39]	384×384	23M	16.5G	27.134ms
TransFG [9]	448×448	86M	108.4G	57.944ms
Original Resnet50	448×448	24M	16.5G	4.174ms
Resnet50-based CMAL-Net (RAW)	448×448	43M	37.4G	6.527ms
Resnet50-based CMAL-Net (RAW+$\mathcal{A}_{oval}$)	448×448	43M	74.8G	12.680ms

Food-11: The proposed approach largely exceeds the previous state-of-the-art approaches with all the three backbone CNNs. The CMAL-Nets based on ResNet50, Res2NeXt50, and TResNet-L beat the second-best method [42] by 3.4%, 3.6%, and 2.5% in terms of accuracy, respectively.

Overall, our approach clearly surpasses the recent state-of-the-art approaches on all three datasets. The best accuracy achieved by our approach on different datasets is reached by different backbones, which is because different original backbones inherently behave differently on different datasets (as shown in Table 4). Using ResNet50 as the backbone, our approach surpasses the recent state-of-the-art approaches on FGVC-Aircraft and Food-11 by a clear margin, and slightly exceeds the recent state-of-the-art approaches on Stanford Cars. Using TResNet-L as the backbone, our approach largely surpasses the recent state-of-the-art approaches on Stanford Cars, which, as mentioned above, demonstrates that our approach can still have a strong enhancing effect on the network architecture even if it is intrinsically able to handle images of a certain domain very well. Moreover, note that, as shown in Table 4, for all the backbones listed in the table, our approach is able to improve the accuracy steadily and significantly.

Following most prior FGVC work, we cited the best accuracy reported in the original papers in Tables 7 and 8 and compared them with our accuracy. We also noted that the previous state-of-the-art methods hardly used a cosine scheduler [23], but generally used a multi-step scheduler. The multi-step scheduler decreases the learning rate step by step by a certain factor for every certain number of epochs. In comparison, the cosine scheduler decreases the initial learning rate relatively rapidly to a minimum value and then rapidly increases the learning rate. The resetting of the learning rate is repeated throughout the training and acts like a simulated restart of the learning process. Such repeated restart facilitates the model to find the global optima.

To investigate the accuracy in similar cases, we tried to adopt the same training strategy as Wang *et al.* [33] (i.e., a multi-step scheduler with an initial learning rate of 0.001 and multiplied by 0.1 after 60 epochs) to train the CMAL-Net built from the same backbone as [33] (i.e., ResNet50). By doing so, CMAL-Net reached 94.2%, 94.8%, and 95.8% accuracy on FGVC-Aircraft, Stanford Cars, and Food-11, respectively. Compared with using a cosine scheduler, the accuracy of CMAL-Net with a multi-step scheduler is slightly decreased by 0.5%, 0.1%, and 0.5% on FGVC-Aircraft, Stanford Cars, and Food-11, respectively. Even so, the accuracy of CMAL-Net on FGVC-Aircraft and Stanford Cars significantly surpasses that of the previous state-of-the-art methods, and the accuracy of CMAL-Net on Stanford Cars is similar to the best accuracy achieved by previous studies. Also, we trained a TResNet-L-based CMAL-Net (which reached the highest accuracy on Stanford Cars in our previous experiments) on Stanford Cars with the above multi-step scheduler and reached 96.8% accuracy. This is a 0.3% drop compared to using cosine scheduler but significantly outperforms the previous state-of-the-art methods on Stanford Cars. These experimental results further illustrate the effectiveness of our method.

To further validate the effectiveness of the proposed framework under the multi-step scheduler, we also trained the original ResNet50 with the training strategy in [33]. To compare with the accuracy of the original ResNet50 with the cosine scheduler in Table 4, we chose FGVC-Aircraft and Stanford Cars as the datasets, on which the accuracies of the original ResNet50 trained with the training strategy in [33] are 90.7% and 92.1%, respectively, falling by 1.6% and 1.0%, respectively, compared to the accuracy in Table 4. From these experimental results, we can observe that (*i*) under the multi-step scheduler, CMAL still improves the accuracy significantly from the original CNN backbone (3.5% and 2.7% on FGVC-Aircraft and Stanford Cars, respectively, which is larger than the improvement under the cosine scheduler), and this implies that the improvement brought in this paper is scheduler-independent; (*ii*) CMAL-Net performs robustly under different schedulers, which is because the proposed MDA provides rich and powerful data augmentations forcing the model to find the global optimum in them. Moreover, a higher baseline always implies a smaller improving space. Under the cosine scheduler, it is difficult for CMAL to achieve as large a boost as under the multi-step scheduler.

4.8. Discussion on the efficiency of CMAL-Net

In this subsection, we discuss the efficiency of the CMAL-Net. For the training phase, each iteration of CMAL-Net is optimized in multiple steps, which obviously increases the training time cost. However, from the perspective of real-world application, training cost is not an important issue, whereas inference cost is crucial. It is because the network weights in real-world applications are trained beforehand, and the network only needs inference for practical use. Therefore, in this subsection, we compare the inference cost of our CMAL-Net with other state-of-the-art methods.

To avoid uncertainties in the implementation for a fair comparison, we select all methods from Table 7 that provide official codes for comparison. The evaluation metrics include model parameters, multiplyaccumulates (MACs), and the time cost of inferring a single image (all the experiments in this subsection are evaluated on Intel Core i7-11700 + NVIDIA GeForce RTX 3080 Ti). The comparison results are shown in Table 9. To achieve the best accuracy, CMAL-Net requires 43M parameters, 74.8G MACs, and 12.680 ms, which is not much compared to the previous state-of-the-art methods. Considering that the accuracy of CMAL-Net significantly exceeds theirs,

CMAL-Net is suitable for applications that require high accuracy and about average inference cost.

5. Conclusion

This paper proposes a novel cross-layer mutual attention learning network (CMAL-Net) for fine-grained visual recognition (FGVC). CMAL-Net utilizes the information learned from shallow and deep layers of a backbone CNN to generate attention regions. The attention region is used as data augmentation to convey the visual clues learned by a specific layer. With this design, we aim to address the problem of intra-class variation and inter-class similarity, together with the difficulty of obtaining large training data. In addition, this design takes advantage of the low-level information of the backbone CNN, which has been ignored in previous FGVC studies. Extensive experimental results show that the proposed CMAL-Net can effectively address the target problems and significantly improve the accuracy of FGVC tasks. The remarkable improvement in classification accuracy is the most significant advantage of CMAL-Net, which clearly surpasses the accuracy of state-of-the-art approaches on three competitive datasets. The weakness is that the multi-step training strategy increases the training time compared to the original backbone networks. However, inference cost is much more critical for real-world applications than training cost. The inference cost of CMAL-Net is not high compared to the previous state-of-the-art approaches but is relatively affordable. This paper shows that mutual learning of deep and shallow layers can substantially improve accuracy and shows how this can be exploited to improve accuracy. For future work, we will research to reduce the cost of training and inference.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Dichao Liu reports financial support was provided by PhD Professional Toryumon Program, Japan. Longjiao Zhao reports financial support was provided by PhD Professional Toryumon Program, Japan.

Data availability

I have shared the link to my code.

References

[1] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, Technical Report, 2013.

[2] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, 4th International IEEE Workshop on 3D Representation and Recognition, 2013. Sydney, Australia

[3] J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net, ICLR (workshop track), 2015.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[6] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, P. Torr, Res2net: A new multi-scale backbone architecture, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2) (2019) 652–662.

[7] T. Ridnik, H. Lawen, A. Noy, E. Ben Baruch, G. Sharir, I. Friedman, Tresnet: high performance gpu-dedicated architecture, in: The IEEE Winter Conference on Applications of Computer Vision, 2021, pp. 1400–1409.

[8] A. Singla, L. Yuan, T. Ebrahimi, Food/non-food image classification and food categorization using pre-trained googlenet model, in: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 2016, pp. 3–11.

[9] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, A. Yuille, Transfg: a transformer architecture for fine-grained recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022.

[10] L. Zhang, S. Huang, W. Liu, Learning sequentially diversified representations for fine-grained categorization, Pattern Recognit. 121 (2022) 108219.

[11] Y. Niu, Y. Jiao, G. Shi, Attention-shift based deep neural network for fine–grained visual categorization, Pattern Recognit. 116 (2021) 107947.

[12] R. Du, D. Chang, A.K. Bhunia, J. Xie, Z. Ma, Y.-Z. Song, J. Guo, Fine-grained visual classification via progressive multi-granularity training of jigsaw patches, in: European Conference on Computer Vision, 2020, pp. 153–168.

[13] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear convolutional neural networks for fine-grained visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1309–1322.

[14] H. Zheng, J. Fu, Z.-J. Zha, J. Luo, Learning deep bilinear transformation for fine–grained image representation, Adv. Neural Inf. Process. Syst. 32 (2019).

[15] Q. Wang, J. Xie, W. Zuo, L. Zhang, P. Li, Deep cnns meet global covariance pooling: Better representation and generalization, IEEE Trans. Pattern Anal. Mach. Intell. 43 (8) (2021) 2582–2597.

[16] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818–833.

[17] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, Y. Wei, Layercam: exploring hierarchical class activation maps for localization, IEEE Trans. Image Process. 30 (2021) 5875–5888.

[18] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Artificial Intelligence and Statistics, 2015, pp. 562–570.

[19] Ö. Çaylı, V. Kılıç, A. Onan, W. Wang, Auxiliary classifier based residual rnn for image captioning, in: 2022 30th European Signal Processing Conference (EUSIPCO), 2022, pp. 1126–1130.

[20] J. Peng, H. Wang, S. Yue, Z. Zhang, Context-aware co-supervision for accurate object detection, Pattern Recognit. 121 (2022) 108199.

[21] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, K. Weinberger, Multi-scale dense networks for resource efficient image classification, in: International Conference on Learning Representations, 2018.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[23] I. Loshchilov, F. Hutter, SGDR: stochastic gradient descent with warm restarts, in: International Conference on Learning Representations, 2017.

[24] Y. Gao, X. Han, X. Wang, W. Huang, M. Scott, Channel interaction networks for fine-grained image categorization, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 10818–10825.

[25] W. Luo, H. Zhang, J. Li, X.-S. Wei, Learning semantically enhanced feature for fine-grained image classification, IEEE Signal Process. Lett. 27 (2020) 1545–1549.

[26] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[27] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: regularization strategy to train strong classifiers with localizable features, in: The IEEE international conference on computer vision, 2019, pp. 6023–6032.

[28] A.F.M.S. Uddin, M.S. Monira, W. Shin, T. Chung, S.-H. Bae, Saliencymix: a saliency guided data augmentation strategy for better regularization, in: International Conference on Learning Representations, 2021.

[29] J. Kim, W. Choo, H. Jeong, H.O. Song, Co-mixup: saliency guided joint mixup with supermodular diversity, in: International Conference on Learning Representations, 2021.

[30] D. LIU, Y. WANG, K. MASE, J. KATO, Recursive multi-scale channel-spatial attention for fine-grained image classification, IEICE Trans. Inf. Syst. 105 (3) (2022) 713–726.

[31] A. Imran, V. Athitsos, Domain adaptive transfer learning on visual attention aware data augmentation for fine-grained visual categorization, in: International Symposium on Visual Computing, 2020, pp. 53–65.

[32] D. Chang, Y. Ding, J. Xie, A.K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, Y.-Z. Song, The devil is in the channels: Mutual-channel loss for fine-grained image classification, IEEE Trans. Image Process. 29 (2020) 4683–4695.

[33] Z. Wang, S. Wang, H. Li, Z. Dou, J. Li, Graph-propagation based correlation learning for weakly supervised fine-grained image classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 12289–12296.

[34] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, Z. Li, Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 9749–9758.

[35] M. Zhou, Y. Bai, W. Zhang, T. Zhao, T. Mei, Look-into-object: self-supervised structure modeling for object recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 11774–11783.

[36] H. Touvron, A. Sablayrolles, M. Douze, M. Cord, H. Jégou, Grafit: learning fine–grained image representations with coarse labels, in: The IEEE International Conference on Computer Vision, 2021, pp. 874–884.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, 2021, pp. 10347–10357.

[38] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, V.N. Boddeti, Neural architecture transfer, IEEE Trans. Pattern Anal. Mach. Intell. 43 (9) (2021) 2971–2989.

[39] M. Chen, H. Peng, J. Fu, H. Ling, Autoformer: searching transformers for visual recognition, in: The IEEE International Conference on Computer Vision, 2021, pp. 12270–12280.

[40] X. Yu, Y. Zhao, Y. Gao, S. Xiong, Maskcov: a random mask covariance network for ultra-fine-grained visual categorization, Pattern Recognit. 119 (2021) 108067.
[41] J. Yao, D. Wang, H. Hu, W. Xing, L. Wang, Adcnn: towards learning adaptive dilation for convolutional neural networks, Pattern Recognition 123 (2022) 108369.
[42] M.T. Islam, B.N.K. Siddique, S. Rahman, T. Jabid, Food image classification with convolutional neural network, in: International Conference on Intelligent Informatics and Biomedical Sciences, volume 3, 2018, pp. 257–262.
[43] P. McAllister, H. Zheng, R. Bond, A. Moorhead, Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets, Comput. Biol. Med. 95 (2018) 217–233.
[44] G. Özsert Yiğit, B.M. Özyildirim, Comparison of convolutional neural network models for food image classification, J. Inf. Telecommun. 2 (3) (2018) 347–357.
[45] K.T. Islam, S. Wijewickrema, M. Pervez, S. O'Leary, An exploration of deep transfer learning for food image classification, in: 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1–5.
[46] A. Şengür, Y. Akbulut, Ü. Budak, Food image classification with deep features, in: 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019, pp. 1–6.
[47] S. Khan, K. Ahmad, T. Ahmad, N. Ahmad, Food items detection and recognition via multiple deep models, J. Electron. Imag. 28 (1) (2019) 013020.
[48] R.Z. Tan, X. Chew, K.W. Khaw, Neural architecture search for lightweight neural network in food recognition, Mathematics 9 (11) (2021) 1245.

Dichao Liu received his B.S. degree from Nanjing University, China, in 2015 and his M.S. and Ph.D. degrees from Nagoya University, Japan, in 2018 and 2022. He is currently a researcher with Navier, Inc., Japan. His current research interests include fine-grained image classification, fine-grained human action recognition, and super-resolution imaging.

Longjiao Zhao received her B.S. degree from University of Science and Technology of China in 2015 and M.S. degree in Information Science, from Nagoya University in 2018. She is currently a Ph.D. Candidate with the Graduate School of Informatics, Nagoya University.

Yu Wang received the M.S. degree in Information Science and Ph.D. degree in Engineering, from Nagoya University, in 2010 and 2013, respectively. He is currently an assistant professor with the College of Information Science and Engineering, Ritsumeikan University.

Jien Kato received the M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. She was a visiting researcher at the University of Oxford from 1999 for one year. She has been a professor at the College of Information Science and Engineering of Ritsumeikan University since 2018.