Inner-Imaging Networks: Put Lenses Into Convolutional Structure

Yang Hu, *Member, IEEE*, Guihua Wen, *Member, IEEE*, Mingnan Luo, Dan Dai, Wenming Cao, *Member, IEEE*, Zhiwen Yu, *Senior Member, IEEE*, and Wendy Hall

***Abstract*—Despite the tremendous success in computer vision, deep convolutional networks suffer from serious computation costs and redundancies. Although previous works address that by enhancing the diversities of filters, they have not considered the complementarity and the completeness of the internal convolutional structure. To respond to this problem, we propose a novel inner-imaging (InI) architecture, which allows relationships between channels to meet the above requirement. Specifically, we organize the channel signal points in groups using convolutional kernels to model both the intragroup and intergroup relationships simultaneously. A convolutional filter is a powerful tool for modeling spatial relations and organizing grouped signals, so the proposed methods map the channel signals onto a pseudoimage, like putting a lens into the internal convolution structure. Consequently, not only is the diversity of channels increased but also the complementarity and completeness can be explicitly enhanced. The proposed architecture is lightweight and easy to be implement. It provides an efficient self-organization strategy for convolutional networks to improve their performance. Extensive experiments are conducted on multiple benchmark datasets, including CIFAR, SVHN, and ImageNet. Experimental results verify the effectiveness of the InI mechanism with the most popular convolutional networks as the backbones.**

***Index Terms*—Channelwise attention, convolutional networks, grouped relationships, inner-imaging (InI).**

Manuscript received April 30, 2020; revised July 21, 2020 and October 7, 2020; accepted October 25, 2020. This work was supported in part by the Guangdong Province Key Area Research and Development Plan Project under Grant 2020B1111120001 and Grant 2018B010107002; in part by the China National Science Foundation under Grant 61273363, Grant 61976092, Grant 61722205, Grant 61751205, Grant 61751202, and Grant U1611461; in part by the Guangzhou Science and Technology Planning Project under Grant 201604020179 and Grant 201803010088; and in part by the Natural Science Foundation of Guangdong under Grant 2018A030313356. This article was recommended by Associate Editor Q. M. J. Wu. *(Corresponding author: Guihua Wen.)*

Yang Hu is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510000, China, and also with the Web Science Institute, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: superhy199148@hotmail.com).

Guihua Wen, Mingnan Luo, Dan Dai, and Zhiwen Yu are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510000, China (e-mail: crghwen@scut.edu.cn).

Wenming Cao is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: wenmincao2-c@my.cityu.edu.hk).

Wendy Hall is with the Web Science Institute, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: wh@ecs.soton.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2020.3034605.

Digital Object Identifier 10.1109/TCYB.2020.3034605

I. Introduction

DEEP convolutional neural networks (CNNs) have exhibited significant effectiveness in modeling image data [1]–[6]; their structures have also been explored continuously [7]–[11]. Meanwhile, CNNs show the bulky size and severe redundancy [12], [13]. Besides pruning a complete structure [14]–[16], lots of methods aim to improve the efficiency of CNNs [17], [18]. Generally, the efficiency heavily depends on the interior components of CNNs, which should meet the following requirements: diversity, complementarity, and completeness. As the basic elements of CNNs, convolutional filters are often modeled to implement channelwise attention [19], which only focuses on improving the diversity of feature maps and lacks explicit modeling of complementarity and completeness of convolution channels.

Some methods have been designed to model the grouping relationship between convolution channels [20]–[22], where Xception [20] encodes the channels modeled by grouping, enhancing the interaction of features between groups; Shufflenet [21] explicitly proposed the interaction and fusion of channels between different groups; and ConDenseNet [22] further verified the excellent performance of convolutional channel-group modeling on the structure of DenseNet [23]. Their outstanding performances indicate that implicit group relations exist between convolutional feature maps. However, the previous works failed to feedback on the modeling of the convolution channel-group relations to the optimization process of the feature maps. On the other hand, the grouping relations are ignored in the ordinary channelwise attention methods since their plain fully connected (FC) encoder cannot represent the grouping and interaction of channel relationships. In other words, these methods have not explicitly modeled the coordination and complementarity between channels.

To overcome the above shortcomings, this article proposes a novel inner-imaging (InI) mechanism, as shown in Fig. 1(b), which is a new way to model the channel relationships. Compared with the channel-relationship modeling in [19] [as shown in Fig. 1(a)], our method first rearranges the feature signals into a pseudoimage $\hat{v} \in \mathbb{R}^{N \times M}$, then it creates grouping filters (G-filters) $w^{(a \times b)}$ to scan on it. In this process, channel signals $u_{ij} \in \hat{v}$ within the same receptive field can build relations from multiple directions (top, down, left, right, top left, top right, bottom left, bottom right, and so on), and are assigned into one group. Subsequently, InI adopts FC layers to model groupwise relationships. That is, the G-filters are responsible for modeling relations between channels in the

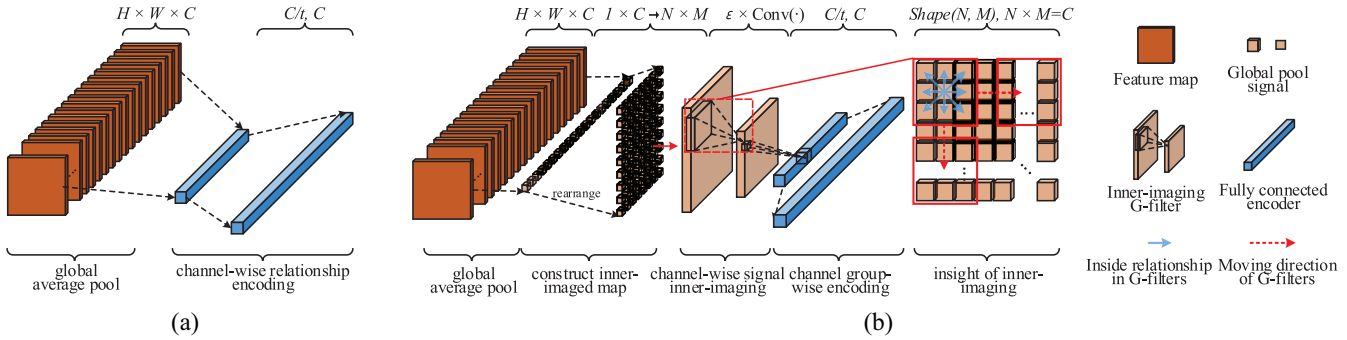

Fig. 1. Architectural comparison of channel relationship modeling. (a) Typical channel relation encoder [19] with considering of identity mapping. (b) InI channelwise relationship modeling architecture, which organizes and models the channel relationships inside each group and between groups.

same group; the followed FC layers are used to model the relationships between groups. With this strategy, the complementarities of both convolutional channels and channel groups are enhanced. On the other hand, the size of channel groups can be flexibly controlled by adjusting the shape of G-filters [such as G-filters $w^{(1\times1)}$ with the shape 1×1 representing the groups with the single channel]. In this way, the completeness of the representation of channels can be improved by integrating multiscale G-filters. The InI-model provides a more complete and precise convolution channel relationship modeling method and provides the channels with more rational rescaling weights.

The exploration of the CNN architecture and modeling of internal network representation is a meaningful and challenging task [24]–[26]. The design of InI brings a new idea of convolution internal structure modeling. It also provides us with a carrier to explore grouping modes of convolution channels. The InI architecture can be applied to all kinds of CNNs to improve the efficiency of CNNs; it is lightweight and easy to implement and understand.

Our contributions can be summarized as follows.

1) A novel InI mechanism is proposed, which first uses G-filters to organize channel signals and simultaneously models the intragroup channel relationships and the intergroup channel relationships. Some theoretical deductions of InI are also provided.
2) The diversity of G-filters with a different utility is explored. Moreover, it is also proposed that multishape G-filters can be integrated to realize the fusion of multisize channel-group modeling.
3) The InI mechanism is employed in some popular CNN structures. This is because it builds inner-imaged maps with both residual and identity mappings, enabling identification flows to participate in the attentional process of residual flows.
4) With the InI mechanism, the effect of diverse channel grouping types is analyzed with the ablation studies for each mode of the InI-model. Besides, the ability of the InI module to collaborate with the spatial attention mechanism [27], [28] is verified.

The remainder of this article is organized as follows. Section II discusses related works. Section III introduces the overall framework of the InI mechanism and its enhanced edition for residual networks (ResNets). Section IV presents our theoretical explanations for designing the InI mechanism. Section V describes the experimental results and analysis. We conclude in Section VI.

II. Related Works

A. Efficient Convolution Structures

Both huge volume and calculation of CNNs [29]–[32] are considered due to their serve redundancy [33], [34], which easily leads to inefficient modeling and overfitting. Some methods impose regularization constraints on the network features [35], [36], or random occlusion or perturbation of intermediate features [37], [38], and some methods attempt to prune the block or channel for an overcomplete convolutional architecture [39]–[41], or use an early exit mechanism [42]. They apply destructive simplification to some complete models, rather than increase the modeling efficiency of finite-scale models. These methods need to build an initial large-scale network and consume computation to optimize pruning operations.

In a parallel line, efficient use of existing components and features is a two-pronged approach [43], [44]. Some methods densely model the feature maps [22], [23], [45]. To make CNN channels organized well, the modeling of channel relationships has attracted research attention [46]. Some approaches attempt to enhance the association of channels [47] and achieve high-efficiency performance by modeling them in the grouping [18], [22]. The studies mentioned above only refine the features of the middle layers repetitively or train the convolution channels in fixed groups, and they do not use the channel relationships to rescale the feature maps. Compared with them, the InI mechanism can model the channel grouping relationships with various sizes and use the attention module to reweight them.

B. Attention and Gating Mechanisms in CNNs

Diversified representation capability is a vital target pursued by machine learning models [48]–[51]. Attention is widely applied to improve the diversity representation of CNNs [52]. It is typically used to model the spatial attentional area [53]–[56] and content meaning [57], including multiscale [58], [59] and multishape [60], [61] features. As a tool for biasing the allocation of resources [19], attention is also used to regulate the internal CNN features [62], [63].

Unlike channel switching, combination [21], [64], channel-wise attention provides an end-to-end training solution for reweighting the intermediate channel features. It can be also combined with spatial attention in various ways, such as juxtaposition [27], sequential [28], or integrated [65].

The above studies either aggregate the features to complement each other or enhance the diversity of the feature maps after a simple encoder. In contrast, the InI design considers both synchronously. We creatively use convolutional filters to organize channel signals on a pseudoimage, like putting lenses in the convolutional networks. This novel strategy reflects the cooperative grouping relations in multiscale and achieves the integrated optimization of the diversity, complementarity, and completeness of CNN channels.

III. Proposed Method

In this section, the overall framework of the InI module is proposed, with a single type of G-filter or combined multi-shape G-filters. Subsequently, the special version of the InI module for ResNets is designed to jointly model the channel signals of residual flow and identity flow.

A. Overall Framework

In each layer of CNNs, each convolutional kernel produces a feature map $\mathbf{u}_l^c \in \mathbb{R}^{W\times H}$, which is defined as a channel. It forms the basic unit of intermediate features in convolutional networks. In order to model the relationships between them, the feature maps $\mathbf{u}_l^c$ are squeezed first, as follows:

$$\mathbf{u}_l^c \in \mathbf{U}_l = F_l(\mathbf{U}_{l-1}, \mathbf{W}_l) = \left[\mathbf{u}_l^1, \mathbf{u}_l^2, \ldots, \mathbf{u}_l^C\right] \tag{1}$$

$$\hat{u}_l^c = F_{sq}(\mathbf{u}_l^c) = AvgGlobalPool(\mathbf{u}_l^c) \tag{2}$$

$$\hat{\mathbf{v}}_{\text{init}} = \left[\hat{u}_l^1, \hat{u}_l^2, \ldots, \hat{u}_l^C\right] \in \mathbb{R}^{1\times C} \tag{3}$$

where $\mathbf{U}_l = [\mathbf{u}_l^1, \mathbf{u}_l^2, \ldots, \mathbf{u}_l^C]$ refers to the feature maps in the layer l and C is the number of channels, $F_l(\cdot)$ is the function of the convolutional layer parameterized by $\mathbf{W}_l$, the function $AvgGlobalPool(*)$ is global average pooling, and $F_{sq}(\cdot)$ denotes the squeeze function with global average pooling. The channel signals $[\hat{u}_l^1, \hat{u}_l^2, \ldots, \hat{u}_l^C]$ can be obtained from feature maps $\mathbf{U}_l$, and the initial channel signal map $\hat{\mathbf{v}}_{\text{init}}$ with the shape of $(1 \times C)$ is also constructed.

Next, by scanning the channel signal map with G-filters $\mathbf{w}^{(a\times b)}$, the channels in the same receptive field with the shape of $(a\times b)$ are organized as a group. However, most G-filters are not available on the initial map $\mathbf{v}_{\text{init}}$ unless $a = 1$. Therefore, a new map, called the inner-imaged map, is generated as

$$\hat{\mathbf{v}}_f = T(\hat{\mathbf{v}}_{\text{init}}) = T\left(\left[\hat{u}_l^1, \hat{u}_l^2, \ldots, \hat{u}_l^C\right]\right) = \begin{bmatrix} \hat{u}_l^{11} & \cdots & \hat{u}_l^{1M} \\ \vdots & \ddots & \vdots \\ \hat{u}_l^{N1} & \cdots & \hat{u}_l^{NM} \end{bmatrix} \in \mathbb{R}^{N\times M}, (N \times M = C) \tag{4}$$

where $\hat{\mathbf{v}}_f$ denotes the inner-imaged map which is folded by the reshape function $T(\cdot)$, its shape is $(N\times M)$. Then, the grouping relations between CNN channels are modeled as follows:

$$\mathbf{v}'[i] = \hat{\mathbf{v}}_f * \mathbf{w}_i^{(a\times b)} = \begin{bmatrix} v_i^{11} & \cdots & v_i^{1m} \\ \vdots & \ddots & \vdots \\ v_i^{n1} & \cdots & v_i^{nm} \end{bmatrix} \tag{5}$$

$$\bar{v}^{xy} = \left(\sum_{i=1}^{\varepsilon}\left(v_i^{xy}\right)\right)\Big/\varepsilon \tag{6}$$

$$\bar{\mathbf{v}}' = \frac{1}{\varepsilon}\sum_{i=1}^{\varepsilon}\left(\mathbf{v}'[i]\right) = \begin{bmatrix} \bar{v}^{11} & \cdots & \bar{v}^{1m} \\ \vdots & \ddots & \vdots \\ \bar{v}^{n1} & \cdots & \bar{v}^{nm} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{7}$$

$$\bar{\mathbf{v}} = \left(F_{\text{flatten}}(\bar{\mathbf{v}}')\right)^\top = \left[\bar{v}^1, \ldots, \bar{v}^{C^g}\right]^\top \in \mathbb{R}^{C^g\times 1} = \left[\bar{v}^{11}, \ldots, \bar{v}^{1m}, \bar{v}^{21}, \ldots, \bar{v}^{2m}, \ldots, \bar{v}^{nm}\right]^\top (n \times m = C^g). \tag{8}$$

In Eq. (5)–(8), the operator $*$ denotes the convolution, and $\mathbf{v}'[i]$ refers to the convolutional result of G-filter $\mathbf{w}_i^{(a\times b)}$ on the inner-imaged map, and its shape is $(n \times m)$, ε is the number of G-filters. The convolutional results are averaged and then applied to obtain the grouping map $\bar{\mathbf{v}}'$ whose element is $\bar{v}^{xy}$. Finally, the grouping map $\bar{\mathbf{v}}'$ is flattened as the tensor $\bar{\mathbf{v}} \in \mathbb{R}^{C^g\times 1}$, where each element of $\bar{\mathbf{v}}$ is a group signal modeled by $\mathbf{w}_i^{(a\times b)}$, and C^g is the number of modeled channel groups.

Obviously, the diversified multishape G-filters can be integrated, such as $\{\mathbf{w}^{(a_1\times b_1)}, \mathbf{w}^{(a_2\times b_2)}, \ldots, \mathbf{w}^{(a_p\times b_p)}\}$, which can be applied to organize the channel groups with different sizes as follows:

$$\mathbf{v}'_j[i] = \left(\hat{\mathbf{v}}_f * \mathbf{w}_i^{(a_j\times b_j)}\right) \in \mathbb{R}^{n_j\times m_j} \tag{9}$$

$$\mathbf{v}'_{1:p}[i] = \left[\mathbf{v}'_1[i] \bowtie \mathbf{v}'_2[i] \bowtie \cdots \bowtie \mathbf{v}'_p[i]\right] \tag{10}$$

$$\bar{\mathbf{v}}'_{1:p} = \left(\frac{1}{\varepsilon}\sum_{i=1}^{\varepsilon}\left(\mathbf{v}'_{1:p}[i]\right)\right) \in \mathbb{R}^{(n_{1:p})\times(m_{1:p})}$$

$$n_{1:p} = \max_{j=1}^{p}\left(n_j\right), m_{1:p} = \sum\nolimits_{j=1}^{p}\left(m_j\right) \tag{11}$$

where $\mathbf{v}'_j[i]$ with the shape of $(n_j \times m_j)$ indicates the convolutional result of the G-filter $\mathbf{w}_i^{(a_j\times b_j)}$ and p is the number of types of G-filter. $\mathbf{v}'_{1:p}[i]$ is the concatenated result of all types G-filters, and $\bowtie$ means matrix concatenation. Since $v'_{1:p}[i]$ is a concatenation of $v'_1[i]$, $v'_2[i]$,..., $v'_p[i]$, the number of columns $m_{1:p}$ of $v'_{1:p}[i]$ is the sum of the number of columns m_j of $v'_1[i]$, $v'_2[i]$,..., $v'_p[i]$. Furthermore, since $\bar{v}'_{1:p}$ is the result of matrix summing of all $v'_{1:p}[i]$, they share the same number of columns $m_{1:p}$. In the case of different shapes of the matrix, zero is automatically filled into the blank. Then, $\bar{\mathbf{v}}'_{1:p}$ is the average result of each ε G-filters with different shapes and sizes. $((n_{1:p}) \times (m_{1:p}))$ is the final shape of the averaged group signal map.

The grouping map is still flatten as

$$\bar{\mathbf{v}} = \left(F_{\text{flatten}}(\bar{\mathbf{v}}'_{1:p})\right)^\top = \left[\bar{v}^1, \ldots, \bar{v}^{C^g}\right]^\top \in \mathbb{R}^{C^g\times 1} \left((n_{1:p}) \times (m_{1:p}) = C^g\right). \tag{12}$$

thus, a greater C^g than that in Eq. (8) is obtained, leading to more samples for the group relationships of CNN channels.

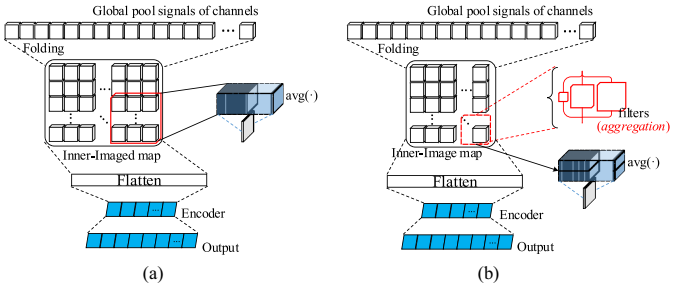

Fig. 2. Detailed structure of the InI. (a) InI module with 3×3 convolutional G-filter. (b) InI module with multishape G-filters aggregation.

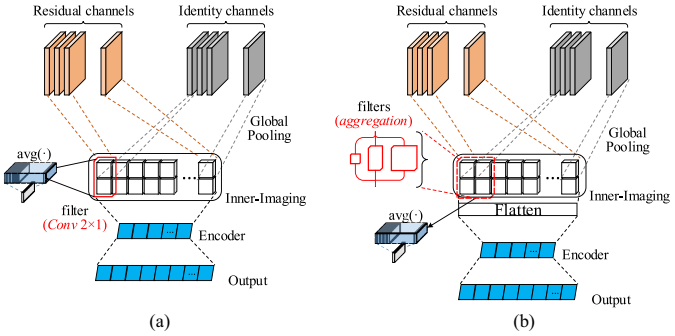

Fig. 3. Special version of simplified InI for ResNets. (a) Simplified InI module with 2×1 convolutional G-filter. (b) Simplified InI module with multishape G-filters aggregation.

After representing the channel groups as $\bar{\mathbf{v}} \in \mathbb{R}^{C^g \times 1}$, the relations are encoded between group signals with FC layers $\mathbf{w}^1 \in \mathbb{R}^{C/t \times C^g}$ and $\mathbf{w}^2 \in \mathbb{R}^{C \times C/t}$. Then, channelwise attention is conducted as follows:

$$\mathbf{W}_{att} = \bigcup_{j=1}^{p} \{\mathbf{w}_1^{(a_j \times b_j)}, \ldots, \mathbf{w}_\varepsilon^{(a_j \times b_j)}\} \cup \{\mathbf{w}^1, \mathbf{w}^2\} \tag{13}$$

$$\mathbf{s} = F_{att}(\mathbf{U}_l, \mathbf{W}_{att}) = \sigma\Big(\mathbf{w}^2 \cdot ReLU\Big(\mathbf{w}^1 \cdot \bar{\mathbf{v}}\Big)\Big) \tag{14}$$

$$\mathbf{U}_l^{att} = \mathbf{s} \circ \mathbf{U}_l = F_{att}(\mathbf{U}_l, \mathbf{W}_{att}) \circ F_l(\mathbf{U}_{l-1}, \mathbf{W}_l) \tag{15}$$

where $\mathbf{W}_{att}$ is the set of parameters in the InI module, which contains all parameters of the G-filters and the FC encoders, $\mathbf{s}$ is the channelwise attentional outputs, F_{att} denotes the summarized function of channelwise attention, operator $\circ$ is the elementswise, product and $\mathbf{U}_l^{att}$ refers to the attentional feature maps. Like all conventional channelwise attention structures, $\mathbf{U}_l^{att}$ refers to the final feature map, which is the product of the original feature map and the attentional value. The number of G-filters ε is set to C/t, where t is the dimensionality-reduction ratio, it is set to 16 as [19].

Fig. 2 shows the detailed structure of the InI module.

B. Special Version for ResNets

In ResNets, the residual flow is considered as a complement to the identity mapping [66], inspired by this argument. We propose the special version of the InI mechanism for ResNets. It attempts to expand the scope of channel relation modeling to both residual and identity channels. This strategy helps residual mappings to supplement identity mappings more precisely.

We define the identity and residual mappings as $\mathbf{X}_l$ and $\mathbf{U}_l$, respectively. The feature map $\mathbf{x}_l^c$ in the identity mappings can be pooled in a similar way to (1). As shown in Fig. 3, the channel signals of $\mathbf{X}_l$ and $\mathbf{U}_l$ can be simply stacked without the operation of folding, as follows:

$$\mathbf{x}_l^c \in \mathbf{X}_l = \left[\mathbf{x}_l^1, \mathbf{x}_l^2, \ldots, \mathbf{x}_l^C\right] \tag{16}$$

$$\hat{x}_l^c = F_{sq}(\mathbf{x}_l^c) = AvgGlobalPool(\mathbf{x}_l^c) \tag{17}$$

$$\hat{\mathbf{x}}_l = \left[\hat{x}_l^1, \hat{x}_l^2, \ldots, \hat{x}_l^C\right] \in \mathbb{R}^{1 \times C} \tag{18}$$

$$\hat{\mathbf{v}}_{\text{stack}} = \begin{bmatrix} \hat{\mathbf{u}}_l \\ \hat{\mathbf{x}}_l \end{bmatrix} = \begin{bmatrix} \hat{u}_l^1 & \hat{u}_l^2 & \cdots & \hat{u}_l^C \\ \hat{x}_l^1 & \hat{x}_l^2 & \cdots & \hat{x}_l^C \end{bmatrix} \in \mathbb{R}^{2 \times C} \tag{19}$$

$$\bar{\mathbf{v}}' = \frac{1}{\varepsilon} \sum_{i=1}^{\varepsilon} \Big(\hat{\mathbf{v}}_{\text{stack}} * \mathbf{w}_i^{(a \times b)}\Big), \quad (a \le 2) \tag{20}$$

$$\bar{\mathbf{v}} = \left\{ \begin{matrix} (\bar{\mathbf{v}}')^\top, & \text{if } a = 2 \\ \left(F_{\text{flatten}}(\bar{\mathbf{v}}')\right)^\top, & \text{if } a < 2 \end{matrix} \right\} \in \mathbb{R}^{C^g \times 1} \tag{21}$$

where $\hat{x}_l^c$ is the squeezed signal of identity feature maps x_l^c, $\hat{\mathbf{v}}_{\text{stack}}$ is the simplified inner-imaged map by stacking the channel signals $\hat{\mathbf{u}}_l$ and $\hat{\mathbf{x}}_l$. Multishape G-filters can also be applied to $\hat{\mathbf{v}}_{\text{stack}}$, as follows:

$$\bar{\mathbf{v}}' = \frac{1}{\varepsilon} \sum_{i=1}^{\varepsilon} \Big(F_{\text{norm}}\Big(\hat{\mathbf{v}}_{\text{stack}} * \mathbf{w}_i^{(a_1 \times b_1)}, \ldots, \hat{\mathbf{v}}_{\text{stack}} * \mathbf{w}_i^{(a_p \times b_p)}\Big)\Big), \quad (\forall a_j : a_j \le 2) \tag{22}$$

$$\bar{\mathbf{v}} = \left\{ \begin{matrix} (\bar{\mathbf{v}}')^\top, & \text{if } \forall a_j : a_j = 2 \\ \left(F_{\text{flatten}}(\bar{\mathbf{v}}')\right)^\top, & \text{if } \exists a_j : a_j < 2. \end{matrix} \right. \tag{23}$$

Although the step of folding the original channel signal map is omitted here, it has a considerable limitation on the shape of group filters: $\forall a : a \le 2$. So, we fold $\hat{\mathbf{v}}_{\text{stack}}$ as

$$\hat{\mathbf{v}}_f = T_{alt}\left(\hat{\mathbf{v}}_{\text{stack}}\right) = \begin{bmatrix} \hat{v}_f^{11} & \cdots & \hat{v}_f^{1M} \\ \vdots & \ddots & \vdots \\ \hat{v}_f^{N1} & \cdots & \hat{v}_f^{NM} \end{bmatrix}$$

$$= T_{alt}\left(\begin{bmatrix} \hat{\mathbf{x}}_l \\ \hat{\mathbf{u}}_l \end{bmatrix}\right) = \begin{bmatrix} \hat{x}_l^1 & \hat{x}_l^2 & \cdots & \hat{x}_l^m \\ \hat{u}_l^1 & \hat{u}_l^2 & \cdots & \hat{u}_l^m \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_l^{\frac{N}{2}1} & \hat{x}_l^{\frac{N}{2}2} & \cdots & \hat{x}_l^{\frac{N}{2}M} \\ \hat{u}_l^{\frac{N}{2}1} & \hat{u}_l^{\frac{N}{2}2} & \cdots & \hat{u}_l^{\frac{N}{2}M} \end{bmatrix}$$

$$\in \mathbb{R}^{N \times M}, (N \times M = 2C) \tag{24}$$

where $\hat{\mathbf{v}}_f$ is the folded inner-imaged map with $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{u}}_l$, and T_{alt} is the alternating reshape function, which enables both residual and identity signals to be scanned in one receptive field.

Fig. 4 shows the structure of the special version InI module on the folded inner-imaged map, with multishape G-filters aggregation. The subsequent process follows (9)–(11); the only difference is that the grouping relations of both residual and identity channels are modeled.

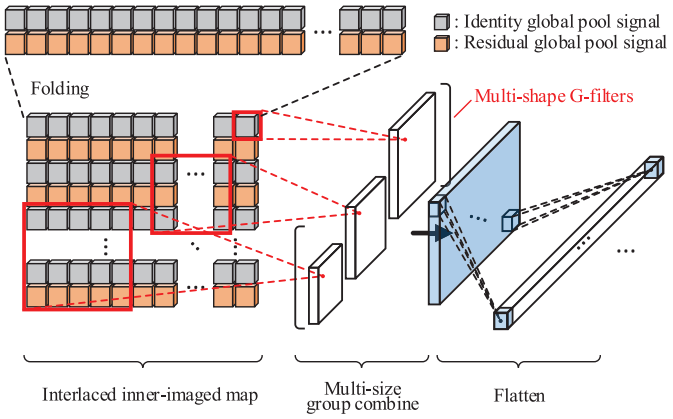

Fig. 4. Folded inner-imaged map of ResNets and the subsequent modeling with multishape G-filters.

FC encoders $\mathbf{w}^1 \in \mathbb{R}^{C/t \times C^g}$ and $\mathbf{w}^2 \in \mathbb{R}^{C \times C/t}$ are still used to model the groupwise relations and output the final channelwise attentional values. As the definition of the residual unit [8]

$$y = \mathbf{X}_l + \mathbf{U}_l = \mathbf{X}_l + F_{res}(\mathbf{X}_l, \mathbf{W}_l) \tag{25}$$

we obtain

$$\mathbf{s} = F_{att}((\mathbf{X}_l, \mathbf{U}_l), \mathbf{W}_{att}) \tag{26}$$

$$\begin{aligned} y &= \mathbf{X}_l + \mathbf{s} \circ \mathbf{U}_l \\ &= \mathbf{X}_l + F_{att}((\mathbf{X}_l, \mathbf{U}_l), \mathbf{W}_{att}) \circ F_{res}(\mathbf{X}_l, \mathbf{W}_l) \end{aligned} \tag{27}$$

where $\mathbf{s}$ refers to the outputs of channelwise attention, and $\mathbf{W}_{att}$ is the total set of parameters in the InI module, which is defined by (13).

IV. Theories

In this section, some theoretical details of the InI model are elaborated and discussed, including its technical advantages and some insightful design motivations.

A. Insight of the Inner-Imaging Mechanism

It is an elegant design method to give new functions and meanings to existing tools. The convolutional filter is usually used to model the spatial features of vision data. Also, it can help us to model the grouping relations of convolutional channels very conveniently.

Compared with the typical channelwise attention mechanism, the InI module can provide more exhaustive and diverse modeling of channel relations, especially on grouping relations.

In this section, we analyze the detailed operation of the InI module and compare it with the conventional channelwise attention. To complete this comparison, we first review the process of original channelwise attention: excluding the output layer $\mathbf{w}^2$, we discuss the FC encoder $\mathbf{w}^1$, as follows:

$$\mathbf{w}^1 = \begin{bmatrix} w_1^{11} & \cdots & w_1^{1C} \\ \vdots & \ddots & \vdots \\ w_1^{\frac{C}{t}1} & \cdots & w_1^{\frac{C}{t}C} \end{bmatrix} \in \mathbb{R}^{\frac{C}{t} \times C} \tag{28}$$

$$\begin{aligned} \mathbf{e} = \mathbf{w}^1 \cdot (\hat{\mathbf{v}}_{\text{init}})^\top &= \begin{bmatrix} w_1^{11} & \cdots & w_1^{1C} \\ \vdots & \ddots & \vdots \\ w_1^{\frac{C}{t}1} & \cdots & w_1^{\frac{C}{t}C} \end{bmatrix} \cdot \begin{bmatrix} \hat{u}_l^1 \\ \vdots \\ \hat{u}_l^C \end{bmatrix} \\ &= \left[\sum_{i=1}^{C} w_1^{1i} \hat{u}_l^i, \ldots, \sum_{i=1}^{C} w_1^{\frac{C}{t}i} \hat{u}_l^i \right] \\ &= \left[e\left(\hat{\mathbf{v}}_{\text{init}}, \mathbf{w}_1^{1[\cdot]}\right), \ldots, e\left(\hat{\mathbf{v}}_{\text{init}} \mathbf{w}_1^{\frac{C}{t}[\cdot]}\right) \right] \in \mathbb{R}^{1 \times \frac{C}{t}} \end{aligned} \tag{29}$$

where $\mathbf{e}$ denotes the embedding of CNN channels, and $e(\cdot, \mathbf{w}_1^{i[\cdot]})$ indicates the feature parameterized by weights $\mathbf{w}_1^{i[\cdot]}$.

In the InI mechanism, there are two successive stages: 1) grouping modeling and 2) FC embedding, as follows:

$$\begin{aligned} \mathbf{s} = F_{att}\Bigg(\mathbf{U}_l, \underbrace{\{\mathbf{w}^{(a_1 \times b_1)}, \ldots, \mathbf{w}^{(a_p \times b_p)}\}}_{a}, \underbrace{\{\mathbf{w}^1, \mathbf{w}^2\}}_{b} \Bigg) \\ \left(\mathbf{w}^{(a_j \times b_j)} = \left[\mathbf{w}_1^{(a_j \times b_j)}, \ldots, \mathbf{w}_\varepsilon^{(a_j \times b_j)} \right] \right). \end{aligned} \tag{30}$$

When the convolutions $\mathbf{w}_1^{(a_j \times b_j)}, \ldots, \mathbf{w}_\varepsilon^{(a_j \times b_j)}$ are abbreviated as

$$\sum_{i=1}^{\varepsilon} \left(\hat{\mathbf{v}}_f * \mathbf{w}_i^{(a_j \times b_j)} \right) \Rightarrow \hat{\mathbf{v}}_f * \mathbf{w}^{(a_j \times b_j)} \tag{31}$$

we obtain

$$\bar{\mathbf{v}}' = \hat{\mathbf{v}}_f * \mathbf{w}^{(a_1 \times b_1)} \bowtie \cdots \bowtie \hat{\mathbf{v}}_f * \mathbf{w}^{(a_p \times b_p)} \tag{32}$$

$$\begin{aligned} \hat{\mathbf{v}}_f * \mathbf{w}^{(a_j \times b_j)} &= \begin{bmatrix} v^{11} & \cdots & v^{1m_j} \\ \vdots & \ddots & \vdots \\ v^{n_j 1} & \cdots & v^{n_j m_j} \end{bmatrix} \in \mathbb{R}^{n_j \times m_j} \\ &= \begin{bmatrix} e_g^j(\hat{\mathbf{g}}_j^1) & \cdots & e_g^j(\hat{\mathbf{g}}_j^{m_j}) \\ \vdots & \ddots & \vdots \\ e_g^j(\hat{\mathbf{g}}_j^{C_g^j - m_j + 1}) & \cdots & e_g^j(\hat{\mathbf{g}}_j^{C_g^j}) \end{bmatrix} \end{aligned} \tag{33}$$

$$\begin{aligned} \mathbf{w}^{(a_j \times b_j)} &= \left[\left[w_j^{11}, \ldots, w_j^{1b_j} \right], \ldots, \left[w_j^{a_j 1}, \ldots, w_j^{a_j b_j} \right] \right] \\ &\Rightarrow \left[w_j^1, \ldots, w_j^{\theta_j} \right], \ (a_j \times b_j = \theta_j) \end{aligned} \tag{34}$$

$$\begin{aligned} \hat{\mathbf{g}}_j^k &= \left[\left[\hat{v}_{\to k}^{11}, \ldots, \hat{v}_{\to k}^{1b_j} \right], \ldots, \left[\hat{v}_{\to k}^{a_j 1}, \ldots, \hat{v}_{\to k}^{a_j b_j} \right] \right] \\ &\Rightarrow \left[\hat{v}_k^1, \hat{v}_k^2, \ldots, \hat{v}_k^{\theta_j} \right] \end{aligned} \tag{35}$$

$$\begin{aligned} e_g^j\left(\hat{\mathbf{g}}_j^k\right) &\Leftarrow e_g\left(\hat{\mathbf{g}}_j^k, \mathbf{w}^{(a_j \times b_j)}\right) \\ &= \sum_{x_k=1}^{a_j} \sum_{y_k=1}^{b_j} w_j^{x_k y_k} \hat{v}_f^{x_k y_k} = \sum_{i=1}^{\theta_j} w_j^i \hat{v}_k^i \end{aligned} \tag{36}$$

where θ_j is the number of parameters in the G-filter $\mathbf{w}^{(a_j \times b_j)}$. $\hat{\mathbf{g}}_j^k$ is the kth receptive field of the G-filter $\mathbf{w}^{(a_j \times b_j)}$, $\to k$ means that the G-filter slides to the kth receptive field. $e_g^j(\cdot)$ is the encoded feature of each channel group.

Retrospect (29), it is noticed that the FC encoder is a particular form in (36), it has only one group which contains all channels.

Next, the role of the FC layer $\mathbf{w}^1$ changes, as follows:

$$\mathbf{e} = \mathbf{w}^1 \cdot \bar{\mathbf{v}} = \mathbf{w}^1 \cdot \left(F_{\text{flatten}}(\bar{\mathbf{v}}')\right)^\top$$

$$= \left[\sum_{k=1}^{C_g} w_1^{1k} e_g\left(\hat{\mathbf{g}}^k\right), \ldots, \sum_{k=1}^{C_g} w_1^{\frac{C}{I}k} e_g\left(\hat{\mathbf{g}}^k\right)\right] \tag{37}$$

$$\Gamma\left(\hat{\mathbf{g}}_j^k\right) \in \left\{(a_j, b_j) : j = 1, 2, \ldots, p\right\} \tag{38}$$

$$\begin{aligned}\bar{\mathbf{v}} &= \left[e_g(\hat{\mathbf{g}}^1), \ldots e_g(\hat{\mathbf{g}}^{C_g})\right] \\ &= \left[e_g^1(\hat{\mathbf{g}}_1^1), \ldots e_g^1(\hat{\mathbf{g}}_1^{C_g^1})\right] \bowtie \cdots \\ &\quad \bowtie \left[e_g^p(\hat{\mathbf{g}}_p^1), \ldots e_g^p(\hat{\mathbf{g}}_p^{C_g^p})\right], \ \left(C_g = \sum_{j=1}^{p} C_g^j\right)\end{aligned} \tag{39}$$

where $\Gamma(\cdot)$ denotes the shape of a matrix, and the one-hot G-filter $\mathbf{o} = [\alpha] \in \mathbb{R}^{1\times 1}$ plays an important role. We obtain

$$\begin{aligned}\hat{\mathbf{v}}_f * \mathbf{o} &= \begin{bmatrix} e_g(\hat{\mathbf{g}}_o^1) & \cdots e_g(\hat{\mathbf{g}}_o^M) \\ \vdots & \ddots \quad \vdots \\ e_g(\hat{\mathbf{g}}_o^{C-M+1}) & \cdots e_g(\hat{\mathbf{g}}_o^C)\end{bmatrix} \\ &\in \mathbb{R}^{N\times M}, (N \times M = C)\end{aligned} \tag{40}$$

$$\hat{\mathbf{g}}_o^k = \begin{bmatrix}\hat{v}^{11} \\ \rightarrow k\end{bmatrix} = \left[\hat{v}_f^{xy}\right], (x \times y = k) \tag{41}$$

$$e_g\left(\hat{\mathbf{g}}_o^k\right) = \alpha \cdot \hat{v}_f^{xy} \propto \hat{u}_l^k \in \hat{\mathbf{v}}_{\text{init}} \tag{42}$$

where $\propto$ means proportional relationship, and $e_g(\hat{\mathbf{g}}_o^k)$ indicates the case of which single channel constructs a group. So

$$\begin{aligned}\sum_{k=1}^{C_g} w_1^{ik} e_g\left(\hat{\mathbf{g}}^k\right) &= \underbrace{\sum_{k=1}^{C} w_1^{ik} e_g\left(\hat{\mathbf{g}}_o^k\right)}_{\text{channels}} + \underbrace{\sum_{k=1}^{C_{\neg o}} w_1^{ik} e_g\left(\hat{\mathbf{g}}_{\neg o}^k\right)}_{\text{groups}} \\ &\quad (C_g - C = C_{\neg o})\end{aligned} \tag{43}$$

where $\neg o$ indicates the non-one-hot G-filter $\mathbf{o}$.

It can be seen that in the two stages of the InI mechanism, *stage-a:* the G-filters generate channel groups in diversified shapes and model channel relations within them, as (32)–(36); *stage-b:* the function of the FC encoder $\mathbf{w}_1$ is applied to modeling the relations between various channel groups, as (37)–(39). The design of the one-hot G-filter $\mathbf{o}$ adds the consideration of modeling the relationship between individual channels and channel groups, as (40)–(43).

It is concluded that the modeled channel relations by the InI mechanism include and much more abundant than that of the typical strategy.

B. Joint Modeling of Residual and Identity Mappings

Another trick is proposed for the InI mechanism in ResNets, which is the joint modeling of residual and identity mappings. It is believed that this trick makes the ResNets more efficient.

As described in [66], the loss ζ is backpropagated (BP) as

$$\begin{aligned}\frac{\partial \zeta}{\partial \mathbf{X}_l} &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\frac{\partial \mathbf{X}_L}{\partial \mathbf{X}_l} \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \frac{\partial}{\partial \mathbf{X}_l}\sum_{i=l}^{L-1} F_{res}(\mathbf{X}_i, \mathbf{W}_l)\right)\end{aligned} \tag{44}$$

where L indicates any deeper residual unit, l indicates any shallower unit, and after using channelwise attention, we obtain

$$\begin{aligned}\frac{\partial \zeta}{\partial \mathbf{X}_l} &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \frac{\partial}{\partial \mathbf{X}_l}\sum_{i=l}^{L-1} F_{att}(\mathbf{U}_i) \right. \\ &\quad \left. \circ F_{res}(\mathbf{X}_i, \mathbf{W}_i)\right) \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(F_{res}(\mathbf{X}_i, \mathbf{W}_i)\frac{\partial F_{att}(\mathbf{U}_i)}{\partial \mathbf{X}_l}\right.\right. \\ &\quad \left.\left. + F_{att}(\mathbf{U}_i)\frac{\partial F_{res}(\mathbf{X}_i, \mathbf{W}_i)}{\partial \mathbf{X}_l}\right)\right)\end{aligned} \tag{45}$$

because $\mathbf{U}_i = F_{res}(\mathbf{X}_i, \mathbf{W}_i)$, we obtain

$$\begin{aligned}\frac{\partial \zeta}{\partial \mathbf{X}_l} &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\mathbf{U}_i\frac{\partial F_{att}(\mathbf{U}_i)}{\partial \mathbf{X}_l}\right.\right. \\ &\quad \left.\left. + F_{att}(\mathbf{U}_i)\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right)\right) \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\mathbf{U}_i\frac{\partial F_{att}(\mathbf{U}_i)}{\partial \mathbf{U}_i}\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right.\right. \\ &\quad \left.\left. + F_{att}(\mathbf{U}_i)\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right)\right)\end{aligned} \tag{46}$$

where

$$\beta = \mathbf{U}_i\frac{\partial F_{att}(\mathbf{U}_i)}{\partial \mathbf{U}_i} + F_{att}(\mathbf{U}_i)$$

we obtain

$$\frac{\partial \zeta}{\partial \mathbf{X}_l} = \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\beta \cdot \frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right)\right). \tag{47}$$

Then, after we set

$$\beta = \mathbf{U}_i\frac{\partial F_{att}(\mathbf{X}_i\mathbf{U}_i)}{\partial \mathbf{U}_i} + F_{att}(\mathbf{U}_i)$$

and

$$\gamma = \mathbf{U}_i\frac{\partial F_{att}(\mathbf{X}_i, \mathbf{U}_i)}{\partial \mathbf{X}_i}$$

with our strategy, we can obtain

$$\begin{aligned}\frac{\partial \zeta}{\partial \mathbf{X}_l} &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \frac{\partial}{\partial \mathbf{X}_l}\sum_{i=l}^{L-1} F_{att}(\mathbf{X}_i, \mathbf{U}_i) \circ F_{res}(\mathbf{X}_i, \mathbf{W}_i)\right) \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(F_{res}(\mathbf{X}_i, \mathbf{W}_i)\frac{\partial F_{att}(\mathbf{X}_i, \mathbf{U}_i)}{\partial \mathbf{X}_l}\right.\right. \\ &\quad \left.\left. + F_{att}(\mathbf{X}_i, \mathbf{U}_i)\frac{\partial F_{res}(\mathbf{X}_i, \mathbf{W}_i)}{\partial \mathbf{X}_l}\right)\right) \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\mathbf{U}_i\frac{\partial F_{att}(\mathbf{X}_i, \mathbf{U}_i)}{\partial \mathbf{X}_l}\right.\right. \\ &\quad \left.\left. + F_{att}(\mathbf{X}_i, \mathbf{U}_i)\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right)\right) \\ &= \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\mathbf{U}_i\left(\frac{\partial F_{att}(\mathbf{X}_i, \mathbf{U}_i)}{\partial \mathbf{U}_i}\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right.\right.\right. \\ &\quad \left.\left.\left. + \frac{\partial F_{att}(\mathbf{X}_i, \mathbf{U}_i)}{\partial \mathbf{X}_i}\frac{\partial \mathbf{X}_i}{\partial \mathbf{X}_l}\right) + F_{att}(\mathbf{X}_i, \mathbf{U}_i)\frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}\right)\right)\end{aligned} \tag{48}$$

so, after sorting out (48), we obtain

$$\frac{\partial \zeta}{\partial \mathbf{X}_l} = \frac{\partial \zeta}{\partial \mathbf{X}_L}\left(1 + \sum_{i=l}^{L-1}\left(\underbrace{\beta \cdot \frac{\partial \mathbf{U}_i}{\partial \mathbf{X}_l}}_{\propto \nabla \mathbf{U}_i} + \underbrace{\gamma \cdot \frac{\partial \mathbf{X}_i}{\partial \mathbf{X}_l}}_{\propto \nabla \mathbf{X}_i}\right)\right) \tag{49}$$

where the gradients of residual and identity mappings are denoted by $\nabla \mathbf{U}_i$ and $\nabla \mathbf{X}_i$. By using the strategy of joint modeling for residual and identity flows, the tradeoff process of identity and residual mappings can be integrated into the BP optimization, so that residual flows can provide more efficient complementary modeling for identity mappings.

V. Experiments

In this section, lots of experiments are conducted to verify the performance of the proposed methods. First, the datasets and implementation details of networks are introduced. Second, the effects of different types of InI are analyzed, while the ablation study for various subassemblies of the InI module is conducted. Third, comparisons of our models with state-of-the-art results are provided. Finally, we give the discussions of experimental results.

A. Datasets

As CIFAR-10, CIFAR-100, SVHN, and ImageNet are often used as benchmark datasets in image recognition experiments. They are also used here.

CIFAR-10 and CIFAR-100 [67]: The two datasets consist of 32×32 colored images. Both of them contain 60 000 images divided equally into 10 and 100 classes. There are 50 000 training images and 10 000 for testing. The standard data augmentation (translation/mirroring) widely adopted as [8] is used for training sets.

SVHN [68]: The street view house numbers dataset contains 32×32 colored images of 73 257 samples in the training set and 26 032 for testing, with 531 131 digits for additional training. Here, all training data are used without data augmentation.

ImageNet [69]: It is used in ILSVRC 2012, which contains 1.2 million training images, 50 000 validation images, and 100 000 for testing, with 1000 classes. Standard data augmentation is adopted for the training set, and the 224×224 crop is randomly sampled. All images are normalized into $[0, 1]$, with mean values and standard deviations.

B. Implementation Details

Networks: The proposed InI module is applied to several popular CNN networks that are taken as backbones, including all convolutional net (All-CNN) [70], preact ResNet [66], wide ResNet (WRN) [32], and pyramidal ResNets (PyramidNets) [71]. All the default settings of the backbones are followed. The InI module is adopted in every block of the backbone networks. For example, ResNet, usually each block contains two convolutional layers. We add the InI module to the last layer of each block. Detailed block settings are described in [32], [66], [70], and [71]. The proposed method is also compared with typical channelwise attention [19] using the same backbones. The batch normalization [72] is performed following G-filters. Our implementation is based on MXNet and GluonCV.[1]

TABLE I
Various Types of G-Filter Sets

Type	Name	Set of G-filters
Square	square-1	$\{(3 \times 3)\}$
	square-2	$\{(1 \times 1), (3 \times 3)\}$
	square-3	$\{(1 \times 1), (3 \times 3), (5 \times 5)\}$
	square-4	$\{(1 \times 1), (2 \times 2), (3 \times 3), (5 \times 5)\}$
	square-5	$\{(1 \times 1), (2 \times 2), (3 \times 3), (4 \times 4), (5 \times 5)\}$
Mix	mix-1	$\{(3 \times 3)\}$
	mix-2	$\{(1 \times 5), (3 \times 3)\}$
	mix-3	$\{(1 \times 5), (3 \times 3), (5 \times 1)\}$
	mix-4	$\{(1 \times 1), (1 \times 5), (3 \times 3), (5 \times 1)\}$
	mix-5	$\{(1 \times 1), (1 \times 5), (3 \times 3), (5 \times 1), (5 \times 5)\}$
Horizontal	horizon-n	$\{(1 \times 1), \ldots, (1 \times n)\}$
Vertical	vertical-n	$\{(1 \times 1), \ldots, (n \times 1)\}$
Simplified[a]	simple-1	$\{(2 \times 1)\}$
	simple-3	$\{(1 \times 1), (2 \times 1), (2 \times 2)\}$
Dilated[b]	d	$\{\ldots, (5 \times 5, s = 2)\}$

[a]Used only for simplified version of InI module in ResNets.
[b]Used only in conjunction with other types of G-filters, not separately.

Training: The SGD with 0.9 Nesterov momentum is used to train models on CIFAR, where epoch number is 300 epochs for ResNet and 200 epochs for other backbones, and 80 epochs on SVHN, where the batch size is 64. For ResNet and All-CNN, the learning rate starts at 0.1 and is divided by 10 at 50%, 75% of the number of total epochs, for WRN, it is divided by 5 at 60, 120, and 160 epochs. On ImageNet, we train models for 100 epochs with the batch size of 256, the initial learning rate is 0.1 and reduced by 10 every 30 epochs.

Settings of the InI Module: Since the size of the inner-imaged map needs to be taken into account in the setting of the G-filter, it follows the rules as if the receptive field of any G-filter exceeds the size of the inner-imaged map in any layer, the G-filter will be automatically discarded.

For the inner-imaged maps, their shapes are defined close to a square: $(n = 20, m = C/10)$ for WRN, $(n = 8, m = C/8)$ for All-CNN, and $(n = 2C/16, m = 16)$ for other backbones. The effects of size changes of the inner-imaged maps are tested on classification performance. For all kinds of ResNet, the special version of the InI module is used by default, which is introduced in Section III-B. It jointly models the identity and residual mappings.

We set the naming rules for the proposed model: 1) the prefix "InI" is used to the name of the model using the InI module and 2) for different G-filter types, the model name is attached with the type name as the suffix, where the type names of G-filters are listed in Table I.

C. Analysis of Different Inner-Imaging Types

Effects of the Shape of G-Filter: As the core component of the InI framework, these issues need to be investigated: whether different G-filter combinations will have a serious impact on model performance (including the number and

[1]https://gluon-cv.mxnet.io

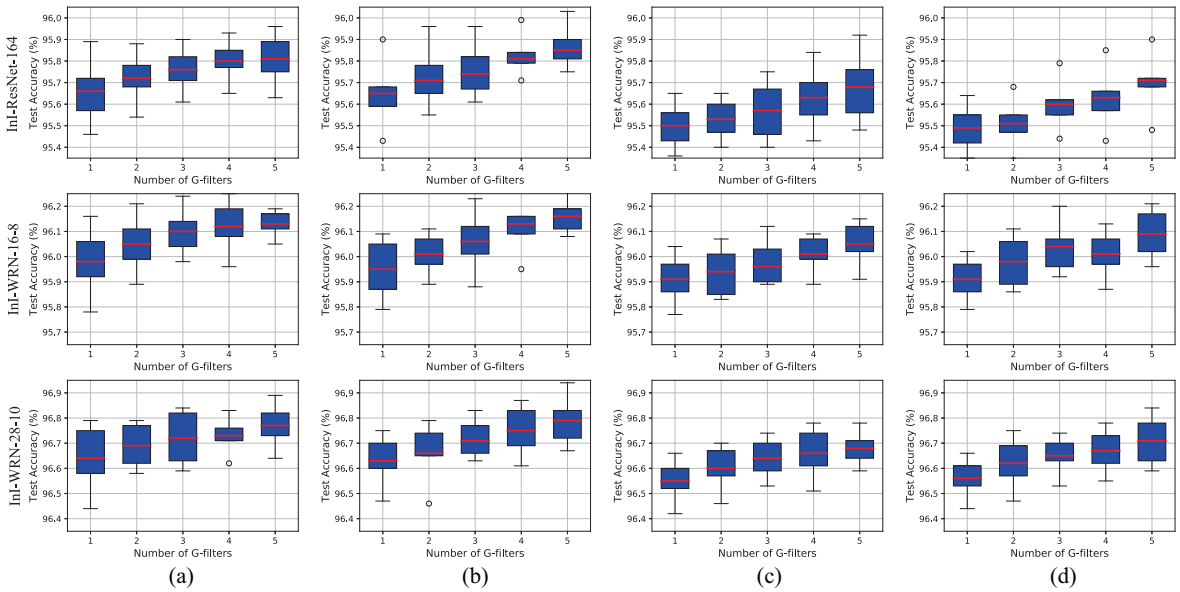

Fig. 5. Test accuracy curves by InI-ResNet and InI-WRN concerning G-filters with various types on CIFAR-10, the results are reported over five runs. For each column, (a) square type, (b) mix type, (c) horizontal type, and (d) vertical type.

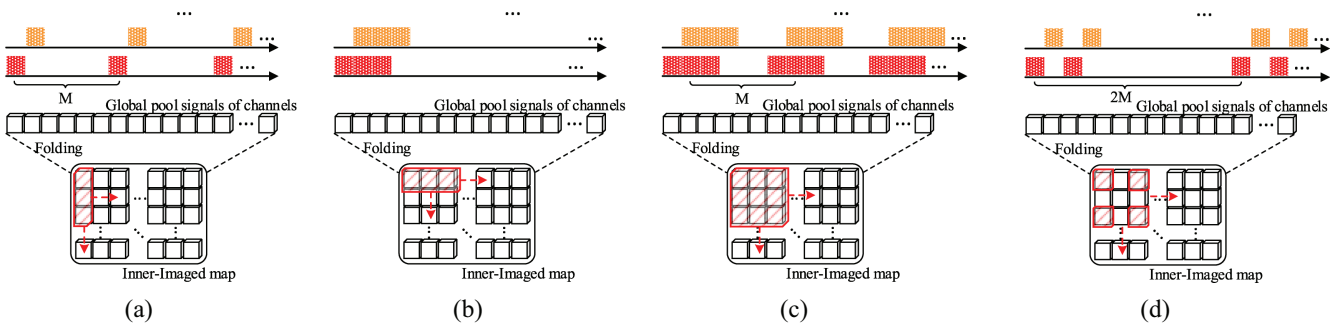

Fig. 6. Examples of channel grouping organization for multiple types of G-filters. (a) G-filter with a vertical shape. (b) G-filter with a horizontal shape. (c) Square G-filter. (d) Dilated G-filter.

shape of the G-filter), and what strategies can be applied to find the better combinations of G-filters.

As listed in Table I, a variety of G-filter combinations is designed, including square, slender (vertical and horizontal), and mixed cases. We also introduce the dilated G-filter, and only one type of the dilated G-filter is used: 5×5 receptive field, dilated rate = 2. The dilated G-filter is used only together with other types of G-filters, not individually. These combinations are applied to the InI models so as to observe their performance changes in different settings, as shown in Fig. 5.

It can be seen from Fig. 5 that the performance of the InI module increases with the number of G-filters. Second, the performance of the types of square and mix G-filters is generally better than that of horizontal and vertical G-filters. Particularly, when the number of square G-filters is relatively large, the performance growth rate tends to be flat, and when the number of mixed G-filters is large, better results can be obtained. These indicate that the diversity of G-filter types can bring benefits to the InI model. Moreover, as deduced in Section IV-A, when we add the (1×1) G-filter to the G-filter set, a more obvious performance improvement can be obtained, which verifies the effect of the special group which only contains one channel, as (49).

Fig. 6 can help us analyze the above results, which illustrates the corresponding channel distribution of groups obtained by using G-filters with various shapes, on the original channel sequence. In the group of channels formed by horizontal and vertical G-filters, the former only contains adjacent channels, while the latter contains channels with slightly distant intervals. These monotonous grouping strategies lead to their mediocre performance. In the channel group formed by the square G-filter, there are adjacent channels and distant channels. However, as the size of the G-filter increases, the coverage of the square G-filter becomes wider and wider, too many square G-filters also reduce the diversity of channel groups. In contrast, the mixed set G-filter composed of horizontal, vertical, and square G-filters is more effective.

It can also be observed in Fig. 6(d) that the dilated [74] G-filter can use fewer parameters than the square G-filter to complete a large scope of channel scanning while overcoming the redundancy caused by a high overlap rate between channel groups. Besides, the analysis of the benefits of the

TABLE II
ERROR ((MEAN ± STD) %) OF ALL-CNN, RESNET, SE-RESNET, AND MULTIPLE MODES OF INI-MODELS OVER FIVE RUNS ON CIFAR-10 AND CIFAR-100. RESULTS THAT SURPASS ALL COMPETING METHODS ARE **BOLD** AND THE OVERALL BEST RESULTS ARE **RED**

Model	Joint	Aggregation	Fold	Dilated	Params.	CIFAR-10	CIFAR-100
All-CNN [70]	–	–	–	–	1.30M	7.25	33.71
SE-All-CNN [19]	–	–	–	–	1.35M	6.55 ± 0.14	32.83 ± 0.21
InI-All-CNN-square-1 (ours)	–	–	√	–	1.35M	6.15 ± 0.13	32.15 ± 0.17
InI-All-CNN-square-3 (ours)	–	√	√	–	1.36M	**6.06 ± 0.12**	**32.02 ± 0.16**
ResNet-110 [66]	–	–	–	–	1.70M	6.37	–
ResNet-164 [66]	–	–	–	–	1.70M	5.46	24.33
SE-ResNet-110 [19]	–	–	–	–	1.75M	5.65 ± 0.15	25.79 ± 0.15
SE-ResNet-164 [19]	–	–	–	–	1.95M	4.79 ± 0.16	22.47 ± 0.20
InI-ResNet-110-square-1* (ours)	–	–	√	–	1.70M	5.46 ± 0.10	25.21 ± 0.17
InI-ResNet-110-simple-1 (ours)	√	–	–	–	1.75M	5.35 ± 0.17	25.12 ± 0.08
InI-ResNet-110-simple-3 (ours)	√	√	–	–	1.77M	5.23 ± 0.13	24.96 ± 0.21
InI-ResNet-110-square-1 (ours)	√	–	√	–	1.76M	5.20 ± 0.12	24.93 ± 0.15
InI-ResNet-110-square-3 (ours)	√	√	√	–	1.77M	5.16 ± 0.14	24.87 ± 0.11
InI-ResNet-110-square-3-d (ours)	√	√	√	√	1.77M	**5.11 ± 0.10**	**24.83 ± 0.09**
InI-ResNet-164-square-1* (ours)	–	–	√	–	1.88M	4.59 ± 0.09	22.14 ± 0.17
InI-ResNet-164-simple-1 (ours)	√	–	–	–	1.95M	4.53 ± 0.12	21.78 ± 0.19
InI-ResNet-164-simple-3 (ours)	√	√	–	–	2.02M	4.30 ± 0.14	21.66 ± 0.11
InI-ResNet-164-square-1 (ours)	√	–	√	–	1.99M	4.34 ± 0.14	21.59 ± 0.16
InI-ResNet-164-square-3 (ours)	√	√	√	–	2.02M	4.24 ± 0.10	21.48 ± 0.18
InI-ResNet-164-square-3-d (ours)	√	√	√	√	2.02M	4.15 ± 0.08	21.44 ± 0.15
InI-ResNet-164-mix-5 (ours)	√	√	√	–	2.04M	4.15 ± 0.09	21.33 ± 0.17
InI-ResNet-164-mix-5-d (ours)	√	√	√	√	2.04M	**4.11 ± 0.13**	**21.31 ± 0.13**

∗: without joint modeling of residual and identity mappings in ResNets (same in hereinafter).

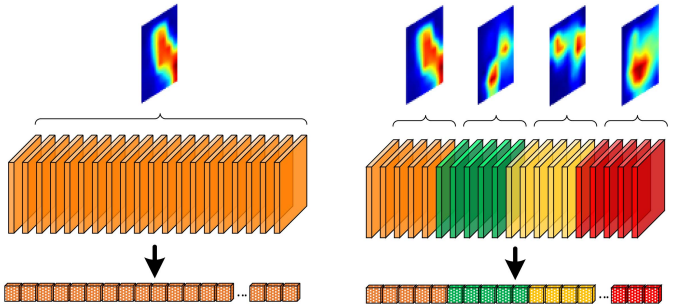

Fig. 7. Illustrations of spatial attention. (a) Single spatial attention. (b) Groupwise multiple spatial attention.

dilated G-filter is given in the appendices in the supplementary material.

Overall, combining G-filters with multiple sizes and shapes can indeed bring better modeling capabilities to the InI mechanisms. The effect on the performance of different G-filter type selection is minimal. Although the maximum performance of InI can be achieved by combining G-filters of various shapes as much as possible, we can obtain competitive performance with the G-filter type "square-3-d." In the appendices in the supplementary material, we also analyze the effects of the shape of inner-imaged map.

D. Ablation Studies

We study the effects of all tricks used in the InI models, which include: 1) jointly modeling from identity and residual mappings (Joint); 2) multishape G-filter aggregation (Aggregation); 3) folded inner-imaged map (Fold); and 4) add dilated G-filter. These tricks are gradually added to the same backbone.

As listed in Tables II and III, the best records with the same backbone are in bold, and the best results are highlighted in red. The variable "Params" in all tables refers to the parameter amount of each model. As the components are gradually added, the test error continues to decrease, and the fully configured InI-models achieved the best results. Besides, many small-scale InI-models achieve performance close to or even better than the larger-scale typical models. Every trick in our InI mechanism shows the improvements in classification results.

It can be found that the well-configured (with three or more proposed tricks) InI models can improve the typical channelwise attention network by nearly 0.6%. Compared with SE-Net, InI-models can bring more performance improvement.

E. Coordination With Spatial Attention

The spatial attention mechanism [28] is a strategy to adjust the pixel-level weights on the feature maps dynamically. In this section, the cooperative ability of the proposed InI mechanism for spatial attention models is validated.

We use the InI module after conducting the spatial attentional operation as follows, and the spatial attention map (SP map) is recorded as ξ

$$\mathbf{U}_l^{\text{new}} = \mathbf{s} \circ F_{spa}(\mathbf{U}_l, \xi) = \left[\xi \circ \mathbf{u}_l^1, \ldots, \xi \circ \mathbf{u}_l^C\right] \tag{50}$$

where $F_{spa}(\cdot)$ is the function of spatial attention and $\mathbf{U}_l^{\text{new}}$ is the final feature matrix in layer l.

As shown in Fig. 7, multiple SP maps $\{\xi^1, \ldots, \xi^\tau\}$ are also introduced on several divided channel groups to highlight the function of the InI model, as

$$\begin{aligned}\mathbf{U}_l^{\text{new}} &= \mathbf{s} \circ F_{spa}\left(\mathbf{U}_l, \{\xi^1, \ldots, \xi^\tau\}\right) \\ &= \left[\xi^1 \circ \mathbf{u}_l^1, \ldots, \xi^1 \circ \mathbf{u}_l^{\frac{C}{\tau}}, \ldots \right. \\ &\quad \left. \xi^\tau \circ \mathbf{u}_l^{C-\frac{C}{\tau}+1}, \ldots, \xi^\tau \circ \mathbf{u}_l^C\right].\end{aligned} \tag{51}$$

TABLE III
TEST ERROR ((MEAN ± STD) %) OF WRN, SE-WRN, AND MULTIPLE MODES OF INI-MODELS OVER FIVE RUNS ON CIFAR-10 AND CIFAR-100. RESULTS THAT SURPASS ALL COMPETING METHODS ARE **BOLD** AND THE OVERALL BEST RESULTS ARE **RED**

Model	Joint	Aggregation	Fold	Dilated	Params.	CIFAR-10	CIFAR-100
WRN-22-10 [32]	–	–	–	–	26.80M	4.44	20.75
WRN-28-10 [32]	–	–	–	–	36.50M	4.17	20.50
SE-WRN-16-8 [19]	–	–	–	–	11.10M	4.58 ± 0.12	20.94 ± 0.17
SE-WRN-22-10 [19]	–	–	–	–	27.05M	4.08 ± 0.13	19.55 ± 0.12
SE-WRN-28-10 [19]	–	–	–	–	36.80M	3.78 ± 0.19	19.03 ± 0.14
InI-WRN-16-8-simple-3 (ours)	√	√	–	–	11.14M	4.01 ± 0.15	19.30 ± 0.11
InI-WRN-16-8-square-1 (ours)	√	–	√	–	11.10M	4.02 ± 0.13	19.29 ± 0.18
InI-WRN-16-8-square-3 (ours)	√	√	√	–	11.14M	3.90 ± 0.09	19.20 ± 0.13
InI-WRN-16-8-square-3-d (ours)	√	√	√	√	11.14M	3.85 ± 0.13	19.16 ± 0.16
InI-WRN-16-8-mix-5 (ours)	√	√	√	–	11.20M	3.84 ± 0.07	19.08 ± 0.19
InI-WRN-16-8-mix-5-d (ours)	√	√	√	√	11.20M	**3.80 ± 0.11**	**19.05 ± 0.15**
InI-WRN-22-10-simple-3 (ours)	√	√	–	–	27.12M	3.63 ± 0.12	18.38 ± 0.13
InI-WRN-22-10-square-1 (ours)	√	–	√	–	27.10M	3.62 ± 0.16	18.43 ± 0.08
InI-WRN-22-10-square-3 (ours)	√	√	√	–	27.12M	3.50 ± 0.18	18.36 ± 0.06
InI-WRN-22-10-square-3-d (ours)	√	√	√	√	27.12M	3.48 ± 0.14	18.27 ± 0.09
InI-WRN-22-10-mix-5 (ours)	√	√	√	–	27.16M	3.51 ± 0.10	18.25 ± 0.13
InI-WRN-22-10-mix-5-d (ours)	√	√	√	√	27.16M	**3.46 ± 0.14**	**18.24 ± 0.09**
InI-WRN-28-10-square-1* (ours)	–	–	√	–	36.60M	3.47 ± 0.11	18.53 ± 0.14
InI-WRN-28-10-simple-1 (ours)	√	–	–	–	36.80M	3.40 ± 0.13	18.30 ± 0.06
InI-WRN-28-10-simple-3 (ours)	√	√	–	–	36.85M	3.33 ± 0.06	18.28 ± 0.04
InI-WRN-28-10-square-1 (ours)	√	–	√	–	36.82M	3.36 ± 0.13	18.26 ± 0.10
InI-WRN-28-10-square-3 (ours)	√	√	√	–	36.88M	3.28 ± 0.10	18.08 ± 0.12
InI-WRN-28-10-square-3-d (ours)	√	√	√	√	36.88M	3.24 ± 0.04	18.06 ± 0.15
InI-WRN-28-10-mix-5 (ours)	√	√	√	–	36.90M	3.21 ± 0.09	18.02 ± 0.12
InI-WRN-28-10-mix-5-d (ours)	√	√	√	√	36.90M	**3.19 ± 0.10**	**17.98 ± 0.07**

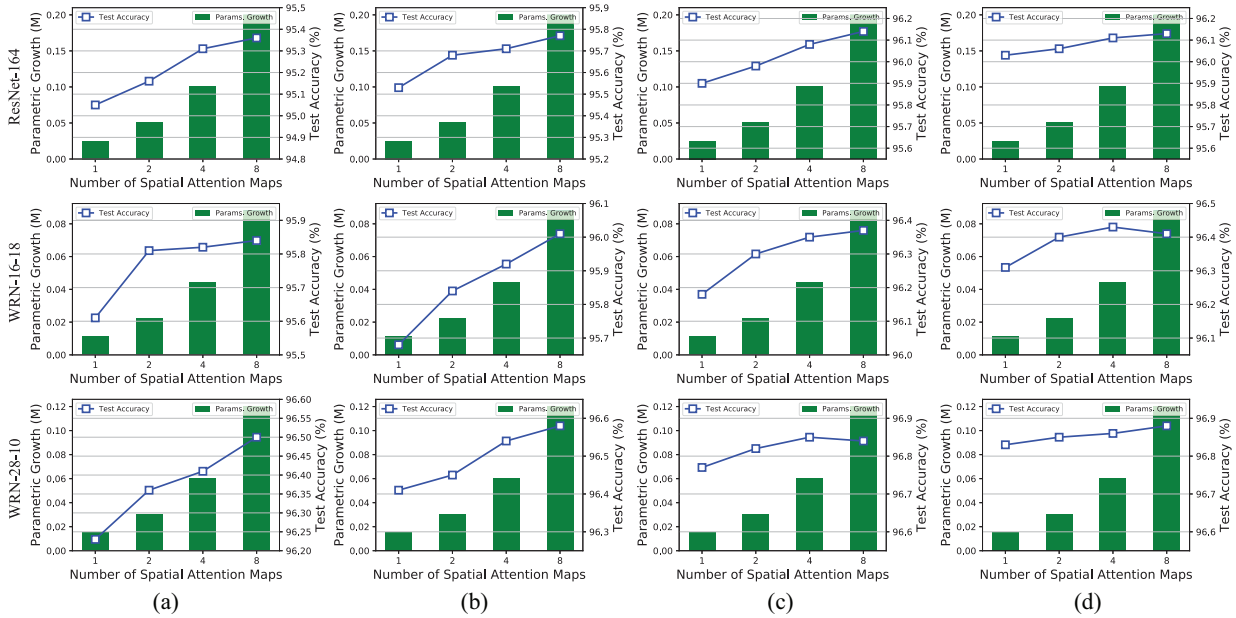

Fig. 8. Amount of parameter growth and test accuracy concerning the number of SP maps with various networks under modes of (a) ordinary, (b) SE, (c) InI-square-3, and (d) InI-mix-5-d on CIFAR-10.

Fig. 8 shows the growth of parameters and test accuracy on CIFAR-10 when using different numbers of SP maps. It is noted that the InI model fits well with the spatial attention mechanism and gets better results than other methods. As the number of SP maps increases, the increments of parameters rise exponentially, and the performance of the model also improves. Still, when the number of SP maps reaches 8, the performance growth becomes less noticeable or even slightly declined.

The possible reason is that too many SP maps lead to a bit overfitting on CIFAR-10 composed with small pictures.

In the appendices in the supplementary material,[2] we provide further comparison results of the proposed InI models with more attention-based models.

F. Comparison Results

Table IV reports the comparison results between the InI-models and state-of-the-art networks, where "FLOPs" denotes the float-point computation amount. "Epoch Time" indicates the average training time per epoch (devices info is given

[2]Also available at https://github.com/scut-aitcm/Appendices-of-InI-Net.

TABLE IV
TEST ERROR((MEAN ± STD) %) OF INI-MODELS OVER FIVE RUNS, COMPARED WITH STATE-OF-THE-ART RESULTS, USING PREACT RESNET, WRN, AND PYRAMIDNET AS BACKBONE, ON CIFAR AND SVHN. THE TOP-THREE RESULTS ARE **RED**, **GREEN**, AND **BLUE**, RESPECTIVELY. THE FLOPS AND EPOCH TIME ARE RECORDED BY RUNNING ON CIFAR-100

Model	Depth	Params.	FLOPs	Epoch Time	CIFAR-10	CIFAR-100	SVHN
original ResNet [8]	110	1.7M	253.1M	37 sec	6.43	25.16	–
pre-act ResNet [66]	164	1.7M	380.6M	65 sec	5.46	24.33	–
	1001	10.2M	2.537G	–	4.62	22.71	–
ResNet Stochastic depth [33]	110	1.7M	–	–	5.23	24.58	1.75
	1202	10.2M	2.840G	–	4.91	–	–
Wide ResNet [32]	28	36.5M	5.243G	112 sec	4.17	20.50	–
FractalNet [13]	21	38.6M	–	–	5.22	23.30	2.01
W/dropout & droppath	21	38.6M	–	–	4.60	23.73	1.87
DenseNet [23]	100	27.2M	13.780G	96 sec	3.74	19.25	1.59
DenseNet-BC ($k = 40$)	190	25.6M	9.388G	–	3.46	17.18	–
ResNeXt [73]	29	34.4M	10.704G	–	3.65	17.77	–
PyramidNet [71]	272	26.0M	8.176G	174 sec	3.31	16.35	–
CliqueNet ($k = 80/k = 150$) [45]	15	8M	6.880G	–	5.17	22.78	1.53
	30	10M	8.490G	–	5.06	21.83	1.64
DMRNet-Wide [43]	32	14.9M	–	–	3.94	19.25	**1.51**
DMRNet-Wide [43]	50	24.8M	–	–	3.57	19.00	1.55
DMRNeXt [43]	29	26.7M	–	–	**3.06**	17.55	–
ConDenseNet [22]	160	3.1M	1.084G	–	3.46	17.55	–
InI-ResNet-simple-3 (ours)	164	2.02M	381.8M	57 sec	4.30	21.66	–
InI-ResNet-square-3 (ours)	164	2.02M	381.9M	59 sec	4.24	21.48	–
InI-WRN-simple-3 (ours)	28	36.85M	5.259G	126 sec	3.33	18.28	–
InI-WRN-square-3 (ours)	28	36.88M	5.262G	128 sec	3.28	18.08	**1.49**
InI-WRN-mix-5 (ours)	28	36.90M	5.266G	131 sec	3.21	18.02	1.53
InI-WRN-mix-5-d (ours)	28	36.90M	5.268G	132 sec	3.19	17.98	**1.47**
InI-PyramidNet-square-3 (ours)	272	27.5M	8.184G	188 sec	3.13	**15.99**	–
InI-PyramidNet-mix-5 (ours)	272	27.8M	8.188G	191 sec	**3.11**	**15.81**	–
InI-PyramidNet-mix-5-d (ours)	272	27.8M	8.121G	192 sec	**3.07**	**15.77**	–

in the appendices in the supplementary material). We only list the Epoch Time records of comparison models we have repeatedly implemented. It can be noticed that the InI-model obviously improves the performance of the baseline method, and the InI-PyramidNet-mix-5-d outperforms state-of-the-art results and achieves the best results on CIFAR-100 and achieves the second-best performance on CIFAR-10. Although the results of the InI-PyramidNet-square-3 are slightly lower, they achieve the third-best and second-best results on CIFAR-10 and CIFAR-100. The InI-WRN-square-3, InI-WRN-mix-5, and InI-WRN-mix-5-d are also very competitive, and two of them meet the best and second-best performance on SVHN. Also, we note that DMRNeXt enhances the interaction and complementarity of the residual flow and the constant flow on the two channels. This scheme of forcibly grouping convolution channels and improving communication between groups has achieved the best performance on the CIFAR-10 dataset after combining with ResNeXt. However, on the more challenging datasets, the proposed InI-Net all performed the best records.

The proposed InI-models have a general and relatively significant performance improvement for the corresponding backbone networks, requiring only little additional parameters and computations. The InI-model has a better convolutional component organization; it makes convolution kernels own the diversity, complementarity, and overall completeness, thus avoiding the redundancy of feature maps and enhancing the integrity of feature representation.

The experimental results on ImageNet are listed in Table V, and the best results are highlighted in bold. It can be seen that InI-ResNet-50-mix-5-d improves the top-1 and top-5 error rate than SE-ResNet-50 by 1.21% and 0.85%, respectively. InI-ResNet-101-mix-5-d exceeds SE-ResNet-101 by 1.04% and 0.8%, respectively, which is even better than the performance of SE-ResNet-152. Compared the networks with mechanisms of both channelwise attention and spatial attention, the InI-ResNet-mix-5-d without spatial attention strategy can obtain better performance than CBAM-ResNet with spatial attention. The results of our InI-ResNet-mix-5-d + spa × 4 improve that of CBAM-ResNet by 0.9%–1.5% on top-1 and 0.6%–0.9% on top-5. Similarly, the smaller InI-models still defeat the larger compared models. The details about the computation and time cost of the experimental models are put in the appendices in the supplementary material.

Fig. 9 illustrates some thermal visualization results of feature maps on ImageNet with different models. It can be found that the InI model can focus more accurately and entirely on recognition objects.

TABLE V
SINGLE CROP ERROR RATES (%) ON IMAGENET

Model	Params.	top-1	top-5
ResNet-50 [8]	25.60M	24.70	7.80
ResNet-101 [8]	44.60M	23.60	7.10
ResNet-152 [8]	60.30M	23.00	6.7
DenseNet-121 [23]	7.98M	25.02	7.71
CliqueNet [45]	14.38M	24.01	7.15
SE-ResNet-50 [19]	28.10M	23.29	6.62
SE-ResNet-101 [19]	49.40M	22.38	6.07
SE-ResNet-152 [19]	65.50M	21.57	5.73
InI-ResNet-50-mix-5-d (ours)	28.77M	22.10	5.79
InI-ResNet-101-mix-5-d (ours)	50.58M	**21.33**	**5.28**
CBAM-ResNet-50 [28]	28.16M	22.66	6.31
CBAM-ResNet-101 [28]	49.48M	21.51	5.69
InI-ResNet-50-mix-5-d + spa (ours)	28.83M	21.44	5.57
InI-ResNet-50-mix-5-d + spa×4 (ours)	29.00M	21.16	5.42
InI-ResNet-101-mix-5-d + spa (ours)	50.66M	20.81	5.17
InI-ResNet-101-mix-5-d + spa×4 (ours)	50.89M	**20.52**	**5.06**

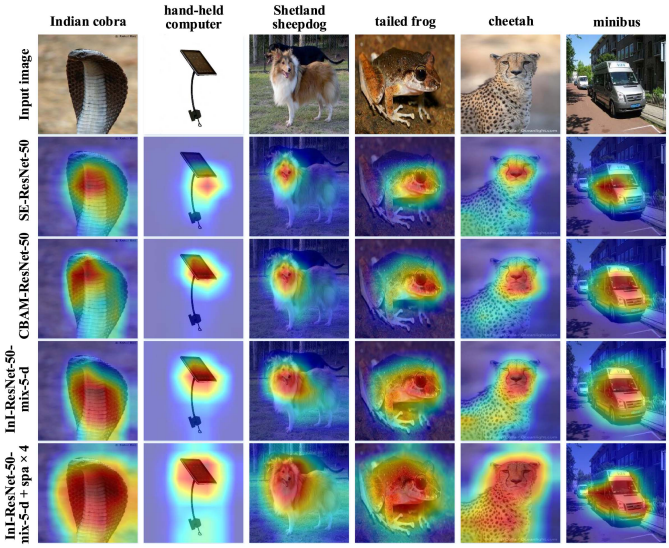

Fig. 9. Grad-CAM [75] visualization for different models with the backbone of ResNet-50, on ImageNet.

The above results on ImageNet show the InI model also works well in large-scale image recognition tasks. In the appendices in the supplementary material, we also show the experimental results of the InI models on the object detection task.

G. Discussion

Our experimental results show that the InI model can improve the modeling ability of CNN, especially for smaller convolution structures. The InI mechanism can adequately stimulate their potential and make the classification results close to or even exceed the original larger CNN models. Besides, the InI model only needs very few additional parameters.

The performance improvement of the InI model than the traditional channelwise attention validates the necessity of channel group relational modeling and the superior modeling ability of the InI model for diversified channel relationships.

In experiments, many possibilities for the implementation of the InI mechanism are discussed. However, it does not mean that the InI model needs to rely on many hyperparametric adjustments. On the contrary, the InI model is insensitive to hyperparameters, such as the G-filter type. Conventional G-filter selection and the simple combination can obtain very competitive performance, illustrating that the InI model is very robust. Also, the InI model is highly scalable and flexible. The various patterns and combinations of the InI model can be extended to pursue extreme excelsior modeling performance.

The InI model has high universality so that it can be applied to any CNN structure. It also has good adaptability to other enhancement mechanisms for CNNs, such as spatial attention.

VI. Conclusion

In this article, we propose the InI architecture for convolutional networks, which present a novel strategy to model the channel relationships in CNNs. The proposed InI architecture uses the convolutional G-filter to organize the grouping relations of the channels, explicitly models the channel in-group coordination and the complementary intergroup ties. This design effectively improves the modeling efficiency of convolutional networks. The proposed method is easy to use and extensible, and its superior performance is verified on multiple benchmark datasets.

In future work, we will test the effectiveness of the proposed InI mechanism on more CNN architectures and apply more data augmentation strategies, like Auto Augmentation [76] and Mixup [77], to further improve the image recognition performance of the proposed models. Also, in recent years, neural architecture search (NAS) [78] has become a new trend in the design of neural network structures. We will integrate the InI mechanism into the paradigm of NAS. When searching for the novel CNN architectures, we plan to give CNN models independent detection ability of grouped channel relations and optimize them.

References

[1] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[2] Y. Wei *et al.*, "Cross-modal retrieval with CNN visual features: A new baseline," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 449–460, Feb. 2017.
[3] L. Wu, J.-Z. Cheng, S. Li, B. Lei, T. Wang, and D. Ni, "FUIQA: Fetal ultrasound image quality assessment with deep convolutional networks," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1336–1349, May 2017.
[4] W. Wu, Y. Yin, X. Wang, and D. Xu, "Face detection with different scales based on faster R-CNN," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 4017–4028, Nov. 2019.
[5] X. Li, Y. Zhang, Q. Cui, X. Yi, and Y. Zhang, "Tooth-marked tongue recognition using multiple instance learning and CNN features," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 380–387, Feb. 2019.
[6] Y. Hu, G. Wen, H. Liao, C. Wang, D. Dai, and Z. Yu, "Automatic construction of chinese herbal prescriptions from tongue images using CNNs and auxiliary latent therapy topics," *IEEE Trans. Cybern.*, early access, May 3, 2019, doi: 10.1109/TCYB.2019.2909925.
[7] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.
[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[9] M. Zhang, W. Li, Q. Du, L. Gao, and B. Zhang, "Feature extraction for classification of hyperspectral and lidar data using patch-to-patch CNN," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 100–111, Jan. 2020.
[10] H. Wang, P. Chen, and S. Kwong, "Building correlations between filters in convolutional neural networks," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3218–3229, Oct. 2017.
[11] L. Wang, X. Qian, Y. Zhang, J. Shen, and X. Cao, "Enhancing sketch-based image retrieval by CNN semantic re-ranking," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3330–3342, Jul. 2020.
[12] X. Zhang, Z. Li, C. C. Loy, and D. Lin, "PolyNet: A pursuit of structural diversity in very deep networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 718–726.
[13] G. Larsson, M. Maire, and G. Shakhnarovich, "FractalNet: Ultra-deep neural networks without residuals," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–6.
[14] R. Yu *et al.*, "NISP: Pruning networks using neuron importance score propagation," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 9194–9203.
[15] G. Zhu, J. Wang, P. Wang, Y. Wu, and H. Lu, "Feature distilled tracking," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 440–452, Feb. 2019.
[16] L. Zeng and X. Tian, "Accelerating convolutional neural networks by removing interspatial and interkernel redundancies," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 452–464, Feb. 2020.
[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[18] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[20] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[21] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[22] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "CondenseNet: An efficient densenet using learned group convolutions," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 2752–2761.

[23] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[24] Z. Liao and G. Carneiro, "A deep convolutional neural network module that promotes competition of multiple-size filters," *Pattern Recognit.*, vol. 71, pp. 94–105, Nov. 2017.

[25] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[26] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4092–4101.

[27] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 6298–6306.

[28] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[29] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4730–4743, Oct. 2018.

[30] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.

[31] F. Tung and G. Mori, "Deep neural network compression by in-parallel pruning-quantization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 568–579, Mar. 2020.

[32] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 87.1–87.12.

[33] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 646–661.

[34] A. Veit, M. J. Wilber, and S. J. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 550–558.

[35] Y. Lu, G. Lu, Y. Xu, and B. Zhang, "AAR-CNNs: Auto adaptive regularized convolutional neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2511–2517.

[36] S. Lin, R. Ji, Y. Li, C. Deng, and X. Li, "Towards compact convnets via structure-sparsity regularized filter pruning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access.

[37] M. Wang, C. Luo, R. Hong, J. Tang, and J. Feng, "Beyond object proposals: Random crop pooling for multi-label image recognition," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5678–5688, Dec. 2016.

[38] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10727–10737.

[39] Z. Wu *et al.*, "BlockDrop: Dynamic inference paths in residual networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 8817–8826.

[40] C. Liu *et al.*, "Computation-performance optimization of convolutional neural networks with redundant filter removal," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 5, pp. 1908–1921, May 2019.

[41] J. Luo, H. Zhang, H. Zhou, C. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN filters for a thinner net," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2525–2538, Oct. 2019.

[42] M. Figurnov *et al.*, "Spatially adaptive computation time for residual networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 1790–1799.

[43] L. Zhao *et al.*, "Deep convolutional neural networks with merge-and-run mappings," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3170–3176.

[44] Z. Huo, B. Gu, and H. Huang, "Training neural networks using features replay," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 6660–6669.

[45] Y. Yang, Z. Zhong, T. Shen, and Z. Lin, "Convolutional neural networks with alternately updated clique," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 2413–2422.

[46] J. Adebayo, J. Gilmer, M. C. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 9505–9515.

[47] H. Gao, Z. Wang, and S. Ji, "ChannelNets: Compact and efficient convolutional neural networks via channel-wise convolutions," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 5197–5205.

[48] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 586–594.

[49] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3730–3737.

[50] Z. Yang, Q. Xu, W. Zhang, X. Cao, and Q. Huang, "Split multiplicative multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5147–5160, Oct. 2019.

[51] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Oct. 2018.

[52] T. V. Nguyen, Q. Zhao, and S. Yan, "Attentive systems: A survey," *Int. J. Comput. Vis.*, vol. 126, no. 1, pp. 86–110, 2018.

[53] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 2285–2294.

[54] W. Li, F. Abtahi, Z. Zhu, and L. Yin, "EAC-Net: Deep nets with enhancing and cropping for facial action unit detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 2583–2596, Nov. 2018.

[55] Y. Li, X. Wang, W. Liu, and B. Feng, "Deep attention network for joint hand gesture localization and recognition using static RGB-D images," *Inf. Sci.*, vol. 441, pp. 66–78, May 2018.

[56] X. Zhang, Y. Su, Z. He, X. Liu, and J. Wu, "Medical exam question answering with large-scale reading comprehension," in *Proc. Nat. Conf. Artif. Intell.*, 2018, pp. 5706–5713.

[57] J. Hou, X. Wu, Y. Sun, and Y. Jia, "Content-attention representation by factorized action-scene network for action recognition," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1537–1547, Jun. 2018.

[58] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 3640–3649.

[59] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[60] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[61] H. Tang, B. Xiao, W. Li, and G. Wang, "Pixel convolutional neural network for multi-focus image fusion," *Inf. Sci.*, vols. 433–434, pp. 125–141, Apr. 2018.

[62] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. C. Courville, "FILM: Visual reasoning with a general conditioning layer," in *Proc. Nat. Conf. Artif. Intell.*, 2018, pp. 3942–3951.

[63] M. F. Stollenga, J. Masci, F. J. Gomez, and J. Schmidhuber, "Deep networks with internal selective attention through feedback connections," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3545–3553.

[64] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognit.*, vol. 74, pp. 15–24, Feb. 2018.

[65] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 9401–9411.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[67] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Handbook of Systemic Autoimmune Diseases*, vol. 1. Amsterdam, The Netherlands: Elsevier, 2009.

[68] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2012, pp. 1–9.

[69] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Feifei, "ImageNet: A large-scale hierarchical image database," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[70] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. Int. Conf. Learn. Represent. (Workshop Track)*, 2014, pp. 1–6.

[71] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 6307–6315.
[72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[73] S. Xie, R. B. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
[74] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–6.
[75] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[76] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation policies from data," 2018. [Online]. Available: arXiv:1805.09501.
[77] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[78] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 55, pp. 1–21, 2019.

Yang Hu (Member, IEEE) received the M.A.Eng. degree from the Kunming University of Science and Technology, Kunming, China, in 2016. He is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China.

He is also a Visiting Researcher in University of Southampton, U.K. His research interests include neural network and deep learning, biomedical information processing.

Guihua Wen (Member, IEEE) received the Ph.D. degree in computer science and engineering from the South China University of Technology, Guangzhou, China,

He is currently a Professor and the Doctoral Supervisor with the School of Computer Science and Technology, South China University of Technology, where he is also a Professor in Chief of the Data Mining and Machine Learning Laboratory. His research area includes cognitive affective computing, machine learning, and data mining.

Mingnan Luo is currently pursuing the master's degree with the College of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His main research interests include image processing and deep learning.

Dan Dai received the M.A.Eng. degree from the Kunming University of Science and Technology, Kunming, China, in 2016. She is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China.

Her research interests include machine learning and biomedical information processing.

Wenming Cao (Member, IEEE) received the M.S. degree in automation from Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree in computer science from City University of Hong Kong, Hong Kong.

His current research interests include data mining and machine learning.

Zhiwen Yu (Senior Member, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008.

He has published more than 100 referred journal papers and international conference papers, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON ENERGY CONVERSION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON MULTIMEDIA, TCVST, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, TNB, *Journal of the International Neuropsychological Society*, PR, *Bioinformatics*, and SIGKDD. His research areas focus on data mining, machine learning, bioinformatics, and pattern recognition.

Dr. Yu is a Senior Member of ACM, IRSS, the China Computer Federation, and the Chinese Association for Artificial Intelligence.

Wendy Hall received the B.Sc. degree and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 1974 and 1977, respectively.

She is a Regius Professor of Computer Science with the University of Southampton, Southampton, U.K., and the Executive Director of the Web Science Institute. She became a Dame Commander of the British Empire in the 2009 U.K. New Year's Honors list.

Dr. Hall is also the Co-Chair of the U.K. Government's AI Review. She has been the President of the ACM, a Senior Vice President of the Royal Academy of Engineering, and a member of the U.K. Prime Minister's Council for Science and Technology. She is a Fellow of the Royal Society and the Royal Academy of Engineering.