Dual Skipping Networks

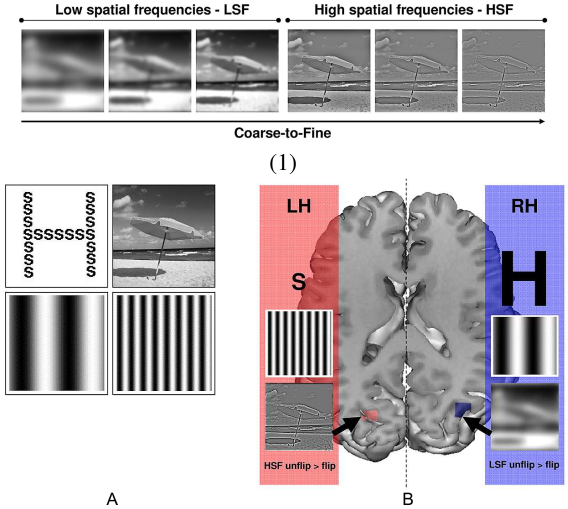
Changmao Cheng[1]*, Yanwei Fu[2]*, Yu-Gang Jiang[1]†,
Wei Liu[3], Wenlian Lu[4], Jianfeng Feng[4], Xiangyang Xue[1]
[1] School of Computer Science, Fudan University [3] Tencent AI Lab
[2] School of Data Science, Fudan University [4] ISTBI, Fudan University

Abstract

Inspired by the recent neuroscience studies on the left-right asymmetry of the human brain in processing low and high spatial frequency information, this paper introduces a dual skipping network which carries out coarse-to-fine object categorization. Such a network has two branches to simultaneously deal with both coarse and fine-grained classification tasks. Specifically, we propose a layer-skipping mechanism that learns a gating network to predict which layers to skip in the testing stage. This layer-skipping mechanism endows the network with good flexibility and capability in practice. Evaluations are conducted on several widely used coarse-to-fine object categorization benchmarks, and promising results are achieved by our proposed network model.

1. Introduction

Though there are still lots of arguments towards the exactly where and how visual analysis is processed within the human brain, there is considerable evidence showing that visual analysis takes place in a predominately and default coarse-to-fine sequence [16] as shown in Fig. 1(1). The coarse-to-fine perception is also proportional to the length of the cerebral circuit path, i.e. time. For example, when the image of Fig. 1(1) is very quickly shown to a person, only very coarse visual stimuli can be perceived, such as sand and umbrella, which is usually of low spatial frequencies. Nevertheless, given a longer duration, fine-grained details with relatively higher spatial frequencies can be perceived. It is natural to ask whether our network has such a mechanism of predicting coarse information with short paths, and fine visual stimuli with longer paths.



(1)

(2)

Figure 1. (1) The coarse-to-fine sequence. (2-A) Example images to assess cerebral asymmetries for spatial frequencies. (2-B) Hemispheric specialization: the left hemisphere (LH)/the right hemisphere (RH) are predominantly involved in the local/global letter identification. Figures from [16].

Another question is how the coarse-to-fine sequence is processed in the human brain? Recent biological experiments [16, 17, 34] reveal that functions of the two cerebral hemispheres are not exactly the same in processing of spatial frequency information. The left hemisphere (LH) and the right hemisphere (RH) are predominantly involved in the high and low spatial frequency processing respectively. As illustrated in Fig. 1(2-A), the top-left figure is a large global letter made up of small local letters, called a Navon figure; the top-right figure is a scene image. The bottom-left and bottom-right are sinusoidal gratings for the images above. In Fig. 1(2-B), given the same input visual stimuli, the highly activated regions in LH and RH are corresponding to high (in red) and low (in blue) spatial frequencies. Additionally, from the view of the evolution process, the left-right asymmetry of the brain structure may be mostly caused by the long-term asymmetrical functions per-

*Equal contribution. Email: {cmcheng16,yanweifu}@fudan.edu.cn
†Corresponding author. Email: ygj@fudan.edu.cn

formed [28].

Some researchers [12] believe that biological plausibility can be used as a guide to design intelligent systems. In the light of this understanding and to mimic the hemispheric specialization, we propose a dual skipping network, which is a left-right asymmetric layer skippable network. Our network can enable the coarse-to-fine object categorization in a single framework. The whole network is structured in Fig. 2. Our network has two branches by referring to LH and RH respectively. Both branches have roughly the same initialized layers and structures. The networks are built by stacking skip-dense blocks, namely groups of densely connected convolutional layers which can be dropped dynamically. The unique connections are built by learning varying knowledge or abstraction. Transition layers aim at manipulating the capacity of features learned from preceding layers. The functionality of each branch is "memorized" from the given input and supervised information in the learning stage. The "Guide" arrow refers to a top-down facilitation of recognition that feeds the high-level information from the coarse branch to relatively lower-level visual processing modules of the fine branch inspired by a similar mechanism in the brain [16, 32, 6]. Though spatial frequency cannot be equated with the granularity of recognition, the dual skipping network might work similar to hemispheric specialization depending on the granularity of supervised information.

The proposed *layer-skipping mechanism* for a single input is to utilize only a part of layers in the deep model for the purpose of computation sparsity and flexibility. The organisms like humans tend to use their energy "wisely" for the recognition and categorization task given visual stimuli [28]. Some recent studies [40] in neuroscience also showed that the synaptic cross-layer connectivity is common in the human neural system, especially in the same abstraction level. In contrast, classical deep convolutional neural networks (e.g. AlexNet [8], VGG [35]) do not have this mechanism and have to run the entire network at inference time. On the other hand, the recent study [5] found that most of data samples are easy to be correctly classified without the utilization of very deep networks. In particular, we propose an affiliated gating network that learns to predict whether skipping several convolutional layers in the testing stage. Our networks are evaluated on three datasets in the coarse-to-fine object recognition tasks. The results show the effectiveness of the proposed network.

Contributions. In this paper, inspired by the left-right asymmetry of the brain, we propose a dual skipping network. The novelties come from several points: (1) The left and right branch network structures towards solving coarse-to-fine classification are proposed. Our network is inspired by the recent theory in neuroscience [16]. (2) A novel *layer-skipping mechanism* is introduced to skip some layers at the testing stage. (3) We employ the top-down feedback facilitation to guide the fine-grained classification via high-level global semantic information. (4) Additionally, we create a novel dataset named small-big MNIST (sb-MNIST) dataset with the hope of facilitating the research on this topic.

2. Related Work

Hemispheric Specialization. Left-right asymmetry of the brain has been widely studied through psychological examination and functional imaging in primates [40]. Recent research showed that both shapes of neurons and density of neurotransmitter receptor expression depend on the laterality of presynaptic origin [17, 34]. There is also a proof in terms of information transfer that indicates the connectivity changes between and within the left and right inferotemporal cortexes as a result of recognition learning [11]. Learning also differs in both local and population as well as theta-nested gamma frequency oscillations in both hemispheres and there is greater synchronization of theta across electrodes in the right IT than the left IT [18]. It has recently been argued that the left hemisphere specializes in controlling routine and tends to focus on local aspects of the stimulus while the right hemisphere specializes in responding to unexpected stimuli and tends to deal with the global environment [41]. For more information, we refer to one recent survey paper [16].

Deep Architectures. Starting with the notable victory of AlexNet [21], ImageNet [8] classification contest has boomed the exploration of deep CNN architectures. Later, [13] proposed deep Residual Networks (ResNets) which mapped lower-layer features into deeper layers by shortcut connections with element-wise addition, making training up to hundreds or even thousands of layers feasible. Prior to this work, Highway Networks [38] devised shortcut connections with input-dependent gating units. Recently, [15] proposed a compact architecture called DenseNet that further integrated shortcut connections to make early layers concatenated to later layers. The simple dense connectivity pattern surprisedly achieves the state-of-the-art accuracy with fewer parameters. Different from the shortcut connections used in DenseNets and ResNets, our layer-skipping mechanism is learned to predict whether skipping one particular layer in the testing stage. The skipping layer mechanism is inspired by the fact that on one-time cognition process, only 1% of total neurons in the human brain are used [28].

Coarse-to-Fine Recognition. The coarse-to-fine recognition process is natural and favored by researchers [46, 47, 29, 7] and it is very useful in real-world applications. Feedback Networks [47] developed a coarse-to-fine representation via recurrent convolutional operations; such that the current iteration's output gives a feedback to the prediction at the next iteration. With the different emphasis on biological mechanisms, our design is also a coarse-to-fine formulation but in a feedforward fashion. Additionally, our coarse-level categorization will guide the fine-level task. HD-CNN [46] learned a hierarchical structure of classes by grouping fine categories into coarse classes and embedded deep CNNs into the category hierarchy for better fine-grained prediction. In contrast, our two-branch model deals with coarse and fine-grained classes simultaneously. Relatedly, research on fine-grained classification [10, 45, 26, 48, 44] has been drawing a lot of attention over the years.

Conditional Computation. Some efforts have been made to bypass the computational cost of deep models in the testing stage, such as network compression [49] and conditional computation (CC) [4]. The CC often refers to the input-dependent activation for neurons or unit blocks, resulting in partial involvement fashion for neural networks. The CC learns to drop some data points or blocks within a feature map and thus it can be taken as an adaptive variant of dropout. [4] introduced Stochastic Times Smooth neurons as binary gates in a deep neural network and termed the straight-through estimator whose gradient is learned by heuristically back-propagating through the threshold function. [1] proposed a 'standout' technique which uses an auxiliary binary belief network to compute the dropout probability for each node. [3] tackled the problem of selectively activating blocks of units via reinforcement learning. Later, [31] used a Recurrent Neural Network (RNN) controller to examine and constrain intermediate activations of a network at test-time. [9] incorporates attention into ResNets for learning an image-dependent early-stop policy in residual units, both in layer level and feature block level. Once stopped, the following layers in the same layer group will not be executed. While our mechanism is to predict whether each one particular layer should be skipped or not, it assigns more selectivity for the forward path.

Compared with conditional computation, our layer-skipping mechanism is different in two points: (1) CC learns to drop out some units in feature maps, whilst our gating network learns to predict whether skipping the layers. (2) CC usually employs reinforcement learning algorithms which have in-differentiable loss functions and need huge computational cost for policy search; in contrast, our gating network is a differentiable function which can be used for individual layers in our network.

3. Model

The presented dual skipping network is overviewed in Fig. 2. The two subsets are shared with the same visual inputs and built upon several types of modules, namely, the shared convolutional layer, skip-dense blocks, transition layers, pooling and classification layers. The motivation and structure of each building block are discussed in this section.

Shared convolutional layer. The input image is firstly processed by the convolutional layer (shared by two following subnets) to extract the low-level visual signal. Such convolutional layers can be biologically corresponding to the primary visual cortex V1 [30]. The two left and right subnets solve the fine-grained and coarse classification respectively.

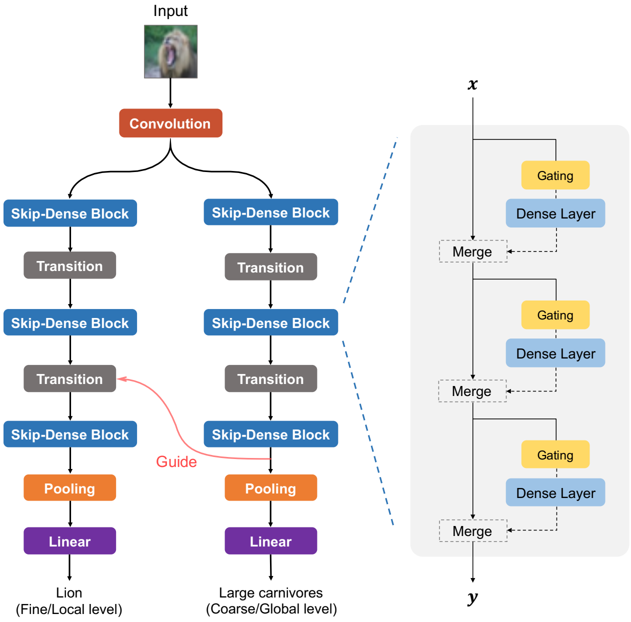


Figure 2. **Dual Skipping Networks: the left-right asymmetric architecture.** (i) Two homogeneous subnets derived from DenseNets process input visual information asymmetrically. One branch is to predict coarse/global level object classes, the other is to predict fine/local level object classes. They share the same preliminary visual processing module, namely the shared convolutional layer. Each subnet is stacked mainly by abstraction blocks and transition layers iteratively. Linear layers transform features into predictions. The top-down guide link delivers feedback information from a high abstraction level of the coarse/global subnet to a lower abstraction level of the fine/local one. (ii) Each skip-dense block contains several skippable densely connected convolutional layers controlled by a cheap affiliated gating network.

Left/right subnet. Each subnet is stacked mainly by skip-dense blocks and transition layers iteratively. As in Fig. 2 (right), we define the skip-dense block which can be viewed as a level of visual concept abstraction; and the gating network is learned whether to block the information flow passing to dense layer. Each transition has convolutional operations with 1 × 1 filter size and pooling, which aims at changing the number and spatial size of feature maps for the next skip-dense block. Both the left and right subnets are almost equivalent in the general structure.

Pooling and classification layers. The global average pooling and linear classifier are added at the top of each branch for the prediction tasks.

3.1. Skip-Dense Block

We give the structure details of a skip-dense block in Fig. 2 (right). For the input pattern $\mathbf{x}$ from a shallower abstraction level, a shortcut connection is applied after each dense layer. And the option of information flowing through each dense layer is checked by a cheap gating network which is input-dependent. The output pattern $\mathbf{y}$ is then processed into the next transition layer and entered into the next abstraction level.

Dense Layer. This type of convolutional layers comes from DenseNet [15] or ResNet [13]. The difference between DenseNet and ResNet comes from the different combination methods of shortcut connections, namely merge functions here. Therefore, the merge operation of preceding layers can be *channel concatenation* or *element-wise addition*, used in DenseNets [15] and ResNets [13] respectively. In particular, the pre-activation unit [14] facilitates both of the merge functions from our observation, thus used in our proposed model. In general, the dense connectivity pattern enables any layer to more easily access proceeding layers and thus use make individual layers additionally supervised from the shorter connections.

Residual networks have been found behaving like ensembling numerous shallower paths [42]. Idling few layers in these networks may not dramatically degrade the performance. Intuitively, the tremendous number of hidden paths may be redundant; an efficient path selection mechanism is essential if we want to reduce the computational cost in the testing stage. Mathematically, there are exponentially many hidden paths in these networks. For example, for L dense layers, we can obtain 2^L hidden paths. Dynamic path selection could pursue the specialized and flexible formations of nested convolution operations. In neuroscience, the experimental evidence [6, 32] also indicates that the path optimization may exist in the parvocellular pathways of the left and right hemispheres when the low and high-pass signals are processed. This supports the motivation of our path selection from the viewpoint of neuroscience.

Gating network. The gating network is introduced for path selection. Particularly, our gating network is learned to judge whether or not skipping the convolutional layer from the training data. It can also be taken as one special type of regularizations: the gating network should be inclined not to skip too many layers if the input data is complex and vice versa. Here, we utilize a $N \times 1$ fully-connected layer for the N-dimensional input features and then a threshold function is applied to the scalar output. We preprocess the input features by average pooling in practice. The parameters of the fully-connected layer are learned from the training set and the threshold function is designed to control the learning process.

Threshold function of gating network. The policy of designing and training the threshold function as an estimator is very critical to the success of the skipping mechanism. Luckily, the magnitude of a unit often determines its importance for the categorization task in CNNs. Based on this observation, we derive a simple end-to-end training scheme for the entire network. Specifically, the output of threshold function is multiplied with each unit of the convolutional layer output, which affects the layer importance of categorization.

The key ingredient of gating network is the threshold function. Given an activation or input, the threshold function needs to judge whether skipping the following dense layer or not as in Fig. 2 (right). Intuitively, the threshold function performs as a binary classifier. In here, we choose hard sigmoid function

$$\mathsf{hard\ sigm}(x) = \max\left(0, \min\left(kx + \frac{1}{2}, 1\right)\right) \qquad (1)$$

for its first derivative in $(0, 1)$ keeps constant, which encourages more flexible path searching compared with the sigmoid function. Besides, for the outputs clipped to 0 or 1, we borrow the idea of straight-through estimator [4] to make the error backpropagated through the threshold function. Thus it is always differentiable.

The slope variable k is the key parameter to determine the output scaling of dense layers. The k is initialized at 1 and increased by a fixed value every epoch. As results, the learned curve of gating will be slope enough to make the outputs of gating network either 0 or 1 in the training process. It can be used as an approximation of step function that emits binary decisions. However, the large slope variable would make the weight training of gating modules unstable. Again, we use the straight-through estimator to keep k equal to one in backward mode.

The mutual adjustment and regularization of gating and dense layers abbreviate the problem of training difficulty and hard convergence of reinforcement learning [3]. For inference, the gating network makes discrete binary decisions to save computation.

3.2. Guide

The faster coarse/global subnet can guide the slower fine/local subnet with global context information of objects in a top-down fashion, inspired from the LSF-based top-down facilitation of recognition in the visual cortex proposed by [2]. In here, we select the output features of the last skip-dense block in the coarse branch to guide the last transition layer in the fine branch. Specifically, the output features are bilinear upsampled and concatenated into the input features of the last transition layer in the local subnet.

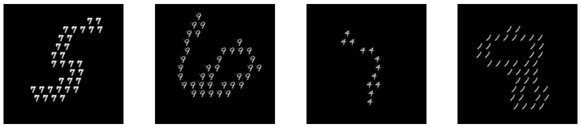


Figure 3. Illustrated examples of sb-MNIST dataset. Each “big” figure is composed of copies of a “small” figure.

The injection of feedback information from the coarse level can be beneficial for the fine-grained object categorization.

4. Experiments

Datasets. We conduct experiments on four datasets, namely sb-MNIST, CIFAR-100 [20] and CUB-200-2011 [43], Stanford Cars [19].

sb-MNIST. Inspired by the experiments with Navon figures in [16], we build sb-MNIST dataset by randomly selecting two images from MNIST dataset and using the first one as the local figure to construct the second one. We generate 12,000 training images and 2,000 testing images for building the dataset. Some examples are illustrated in Fig. 3.

CIFAR-100 has 60,000 images from 100 fine-grained classes, which are further divided into 20 coarse-level classes. The image size is 32 × 32. We use the standard training/testing split[1].

CUB-200-2011 contains 11,788 bird images of 200 fine-grained classes. Strictly following the biological taxonomy, we collect 39 coarse labels in total by the family names of the 200 bird species. For instance, the black-footed albatrosses belong to Diomedeidae family. We use the default training/testing split.

Stanford Cars is another fine-grained classification dataset. It contains 16,185 car images of 196 fine classes (e.g. Tesla Model S Sedan 2012 or Audi S5 Coupe 2012) which describe the properties such as Maker, Model, Year of the car. In terms of the basic car types defined in [47], it can be categorized into 7 coarse classes containing Sedan, SUV, Coupe, Convertible, Pickup, Hatchback and Wagon. The default training/test split is used here.

Competitors. We compare the following several baselines. (1) *Feedback Net* [47]: as the only work of enabling the coarse-to-fine classification, it is a feedback based learning architecture in which each representation is formed in an iterative manner based on the feedback received from previous iteration’s output. The network is instantiated using existing RNNs. Thus Feedback Net has a better classification performance than the standard CNNs. (2) *DenseNet* [15]: DenseNet connects each layer to all of its preceding layers in a feed-forward fashion; and the design of our skip-dense block is derived from DenseNet. Thus comparing with DenseNet, our network has two new components: the gating network to skip some layers dynamically and the two-branch structure for solving the coarse-to-fine classification. (3) *ResNet* [13]: it is an extension of traditional CNNs by learning the residual of each layer to enable the network of being trained substantially deeper than previous CNNs.

Implementation details. The merge types of our models can be *channel concatenation* or *element-wise addition*, “Concat” and “Add” for short. In the experiments, we configure our model based on Concat merge type, namely DenseNets as default. The shared convolution layer with output channels of twice the growth rate is performed to the input visual images. We replace the outputs of skipped dense layers with features maps of the same size filled with zero at inference.

For sb-MNIST, we configure 4 skip-dense blocks each with 3 dense layers and growth rate as 6. We do not use guide link for sb-MNIST dataset. For CIFAR-100, we verify our method with different model settings. Two settings of Concat merge type are based on DenseNet-40 and DenseNet-BC-100 [15], denoted as Concat-40 and Concat-BC-100 respectively. We also report the results of our model with Add merge type. Referred as Add-166, the model is built very similar to ResNet-164 [14], except for Add-166 including two extra transition layers. Data preprocessing procedure and initializations follow [15, 14].

For CUB-200-2011 and Stanford Cars, our model is built on DenseNet-121 [15]. The input images are resized to 360 × 360 for training and test on both datasets. We incorporate the ImageNet pre-trained DenseNet-121 weights into our model as network initialization. Mini-batch size is set to 16, learning rate is started with 0.01. To save GPU memory, we use the memory-efficient implementation of DenseNets [33] here. The experiments are conducted without using any bounding boxes or part annotations.

All of our models are trained using SGD with a cosine annealing learning rate schedule [27] and Nesterov momentum [39] of 0.9 without dampening. The models are trained jointly with or without gating modules, then fine-tuned with gating. We found that joint training as a start speeds up the whole process and makes the performance more stable. We do gradient clipping with L_2 norm threshold 1.0 for avoiding gradient explosion. For the gating modules whose outputs are all 0 or 1 in a batch, we apply auxiliary binary cross entropy loss on them to guarantee functioning of the gating modules. All the code is implemented in PyTorch and run on Linux machines equipped with the NVIDIA GeForce GTX 1080Ti graphics cards.

4.1. Main Results and Discussions

4.1.1 sb-MNIST

The results on this dataset are compared in Tab. 1. The “Local” and “Global” indicate the classification of “small” fig-

[1] `https://www.cs.toronto.edu/~kriz/cifar.html`

Methods	Accuracy (%)	
	Local	Global
LeNet [23]	90.3	70.2
DenseNet	98.3	97.2
ResNet	98.1	96.8
Ours	**99.1**	**99.1**

Table 1. Results on sb-MNIST dataset. Our network has 17 layers and averagely ~20% layers can be skipped in the testing step. DenseNet and ResNet have 40 and 18 layers respectively.

Methods	Accuracy (%)		Params
	Local	Global	
[36]	72.6	–	–
All-CNN [37]	66.3	–	–
Net-in-Net [25]	64.3	–	–
Deeply Sup. Net [24]	65.4	–	–
FractalNet [22]	76.7	–	38.6M
Highway [38]	67.6	–	–
Feedback Net [47]	71.1	80.8	1.9M
DenseNet-40 [15]	75.6	80.9	1.0M×2
DenseNet-BC-100 [15]	77.7	83.0	0.8M×2
ResNet-164 [14]	75.7	81.4	1.7M×2
Concat-40	**75.4**	**82.9**	0.9M+0.7M
Concat-BC-100	**76.9**	**83.4**	0.7M+0.6M
Add-166	**75.8**	**82.5**	1.5M+0.5M

Table 2. Results on CIFAR-100 dataset. The virtual depth of Feedback Net is 48, indicating the number of unfolded layers in Feedback Net, reported from [47]. "×2" indicates two separate branches for two tasks. $a + b$ in the "Params" column represents the averagely used parameters a, b in Local and Global branches of our models. The Global results of DenseNet and ResNet are run by ourselves.

ures and "big" figures. On this dataset, DenseNet, ResNet and LeNet are run separately for the both "Local" and "Global" tasks. The classification of "small" and "big" figures in the image corresponds to the identification task of high and low spatial frequency processing [16]. This toy dataset gives us a general understanding of it.

Judging from the results in Tab. 1, our method outperforms other methods by a clear margin. It is largely due to two reasons. Firstly, the skip-dense blocks can efficiently learn the visual concepts; secondly, the gating network learns to find optimal path routing for avoiding suffering from both overfitting and underfitting. The performance of original version LeNet is greatly suffered from the small number of layers and convolutional filters while DenseNet and ResNet show slight overfitting. Averagely, our model uses 76% of all the parameters as 15% and 31% of dense layers in "Local" and "Global" branches are skipped. Due to a portion of layers skipped in testing step, the total running time and running cost are lower.

4.1.2 CIFAR-100

We compare the results in Tab. 2. We compare Feedback Net, DenseNet and ResNet as well as other previous works. The "Local" and "Global" indicate the classification tasks at fine-grained and coarse levels individually. DenseNet and ResNet are run separately for the "Local" and "Global" tasks respectively.

In Tab. 2, our models outperform several baselines and show comparable performances to the corresponding base models with much less computational cost. Compared with the Feedback Net, Concat-40 gains the accuracy margins of 4.3% and 2.1% on "Local" and "Global" individually with relatively fewer parameters. This implies that our network has the better capability of learning the coarse-to-fine information by the two branches. Though Feedback Net has a shallower physical depth 12, it contains more parameters and consumes more computation cost. This difference is largely caused by the different architectures of two types of networks: Feedback Net is built upon the recurrent neural network, while our network is a forward network with the strategy of skipping some layers. Thus, our network is more efficient in term of computational cost and running time.

On the "Global" task, our network can still achieve a modest improvement over other baselines. One possible explanation is that since the "Global" task is relatively easy, the networks with standard layers may tend to overfit the training data; in contrast, our gating network can skip some layers for better regularization.

4.1.3 CUB-200-2011 and Stanford Cars datasets

We compare the results on CUB-200-2011 and Stanford cars datasets in Tab. 3 and Tab. 4 respectively. The "Local" and "Global" tasks still refer to the fine-grained and coarse-level classification. Our network uses 121 layers and averagely ~30% and ~13% of dense layers of global and local branches can be skipped in the testing step.

On the coarse classification, our result is better than those baselines. On the fine-grained task, our models achieve results comparable to the state-of-the-art [10] with much fewer parameters. On CUB-200-2011 dataset, our result is still better than that of MG-CNN, Bilinear-CNN and RA-CNN (scale 1+2) which employ multiple VGG networks [35] for classification, showing the effectiveness of our compact structure. Comparably, each sub-network of Bilinear-CNN uses the pre-trained VGG network and there is no information flow between two networks until the final fusion layer.

4.2. Ablation Studies on CIFAR-100 dataset

We also conduct some ablation studies to further evaluate and explain the impacts on performances by choosing key settings in our models.

Methods	Accuracy (%)		Params
	Local	Global	
DenseNet-121 [15]	81.9	92.3	8M×2
ResNet-34 [13]	79.3	91.8	22M×2
MG-CNN [44]	81.7	–	≥144M
Bilinear-CNN [26]	84.1	–	≥144M
RA-CNN (scale 1+2) [10]	84.7	–	≥144M
RA-CNN (scale 1+2+3) [10]	**85.3**	–	≥144M
Ours	84.9	**93.4**	6M+5M

Table 3. Results on CUB-200-2011 dataset. The results of DenseNet-121 and ResNet-34 are run by ourselves.

Methods	Accuracy (%)		Params
	Local	Global	
Feedback Net [47]	53.4	87.4	1.5M
DenseNet-121 [15]	90.5	92.8	8M×2
ResNet-34 [13]	89.3	92.0	22M×2
Bilinear-CNN [26]	91.3	–	≥144M
RA-CNN (scale 1+2) [10]	91.8	–	≥144M
RA-CNN (scale 1+2+3) [10]	**92.5**	–	≥144M
Ours	92.0	**93.8**	6M+5M

Table 4. Results on Stanford Cars dataset. The results of DenseNet-121 and ResNet-34 are run by ourselves.

4.2.1 Merge types and gating functions

We compare the different choices of merge types and gating functions. For merge types, we use Concat-BC-100 and Add-166 for Concat and Add types respectively since the two models have roughly the same number of bottleneck dense layers (48 and 54). For gating functions, we compare the proposed hard sigm with soft $\text{sigm}(x) = \frac{1}{1+e^{-x}}$.

The results are compared in Tab. 5. Two evaluation metrics have been used here, namely accuracy and skip ratio. The skip ratio is computed by the number of skipped layers dividing the total number of dense layers in each individual branch.

Judging from the results in Tab. 5, the combination of channel concatenation for merge and gating via hard sigmoid is our best network configuration as it achieves the best performance with the fewest parameters. Besides, we notice that hard sigm is generally better than soft sigm for both merge types on most cases, indicating that the former possesses a stronger ability of routing path searching.

We compare the skip ratios on "Local" and "Global" tasks. In testing stages, the coarse branch of our network will skip much more layers than its sibling fine branch. This is largely due to the fact that the "Global" task is easy, and using relative a few layers of coarse branch is good enough to grasp the coarse-level information. Interestingly, this point follows the coarse-to-fine perception in the recent neural science study [16] that low spatial frequency information is predominantly processed by right hemisphere faster with relatively shorter paths in the cerebral system.

Merge	Gating	Accuracy (%)		Skip Ratio (%)	
		Local	Global	Local	Global
concat	hard	76.9	83.4	15.3	29.7
	soft	73.8	79.7	15.4	30.5
add	hard	75.8	82.5	10.4	39.8
	soft	74.2	83.1	6.1	23.5

Table 5. Ablation study on CIFAR-100 dataset. 'concat', 'add','hard','soft' indicate the channel concatenation, elementwise addition, soft sigmoid function and hard sigmoid function.

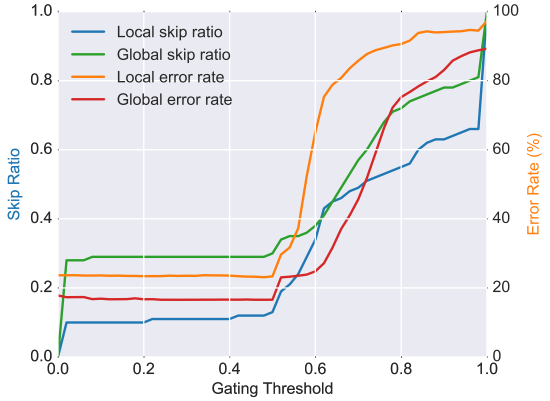


Figure 4. Local and Global layer-skipping ratios and error rates under different gating thresholds. The slope variable is properly scaled for the smooth change of the skip ratios.

4.2.2 Skip ratio vs. error rate

To further verify the flexibility of our model, we take the Concat-BC-100 model with hard sigm for gating as the study object since it achieves the excellent performance with very few parameters. We vary the threshold of gating network which can affect the layer-skipping ratios and error rates of "Local" and "Global" tasks, as reported in Fig. 4.

From the results, we can conclude that both branches are capable of producing acceptable accuracy within a certain range of gating thresholds. As expected, the optimal range of skip ratios for two branches are not same due to the different granularity level of recognition tasks. For the Global branch, $0\% - 35\%$ is the optimal skip ratio range. For the Local branch, $0\% - 20\%$ is the optimal skip ratio range, which is more strict than the Global branch. The good thing is that the learned gating module at each dense layer makes the performance of two branches keep consistent under various user-defined thresholds. As the gating threshold value is raised above 0.5, the skip ratios of two branches increase rapidly and the error rates rise synchronously, meaning that the network becomes disordered and weak in expressivity. We also notice that the performances of two trained branches without or with little layer skipping keep almost undamaged even we didn't train the full network without

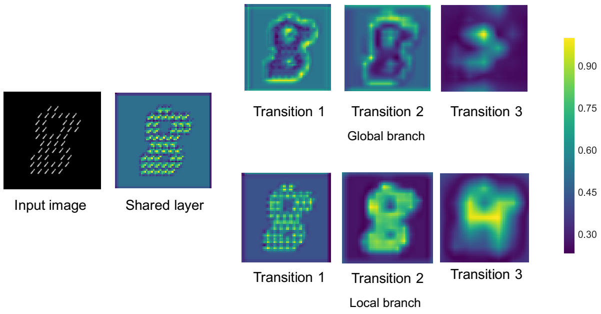


Figure 5. Feature visualization at shared and transition layers for Global and Local branches.

gating when fine-tuning. We conjecture two reasons behind this intriguing phenomenon. First, the shortcut connections make the model very stable and robust to the fine weight updating. The second reason is that there still exist some data samples using the whole dense layers when training with gating.

4.3. Feature Visualization

Investigating the responses of intermediate feature maps at different abstraction levels facilitates our understanding on nueral networks. Especially, the asymmetry of two branches leads to discrepant manipulatations on the given input. Here, we do feature visualization on different abstraction levels. Technically, we extract the output features from the first shared convolutional layer and transition layers of two branches. Then the absolute values of the feature maps are applied over by channel-wise average pooling and scaled to $[0, 1]$ for visualization.

A case study on sb-MNIST dataset is shown in Fig. 5. From the visualization results, we can observe that the first shared features hold both global and local information. Then Global branch cares more about global context information and grasps the shape or style of big figure "8" very quickly at transition layer 1 and 2. Then the transformed features at transition layer 3 of the Global branch represent the semantic information of the big figure. While the Local branch focuses more on the local details of the big figure and neglects the background instantly at transition layer 1. And it keeps the shape of the big figure "8" almost unchanged until the last transition layer, which means the Local branch does not learn the pattern of the big figure. Instead, it transforms the local details, namely the features of small figures, into the final representation.

5. Conclusion

Inspired by the recent study on the hemispheric specialization and coarse-to-fine perception, we proposed a novel left-right asymmetric layer skippable network for coarse-to-fine object categorization. We leveraged a new design philosophy to make this network simultaneously classify coarse and fine-grained classes. In addition, we proposed the layer-skipping behavior of densely connected convolutional layers controlled by an auxiliary gating network. The experiments conducted on three datasets validate the performance, showing the promising results of our proposed network.

Acknowledgement

This work was supported by two projects from NSFC (#61622204 and #61702108) and two projects from STCSM (#$16JC1420401$ and #$16QA1400500$).

References

[1] J. Ba and B. J. Frey. Adaptive dropout for training deep neural networks. In *NIPS*, 2013.

[2] M. Bar, K. S. Kassam, A. S. Ghuman, J. Boshyan, A. M. Schmid, A. M. Dale, M. S. Hämäläinen, K. Marinkovic, D. L. Schacter, B. R. Rosen, et al. Top-down facilitation of visual recognition. *PNAS*, 103(2), 2006.

[3] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup. Conditional computation in neural networks for faster models. *CoRR*, abs/1511.06297, 2015.

[4] Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[5] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama. Adaptive neural networks for efficient inference. In *ICML*, 2017.

[6] J. Bullier. Integrated model of visual processing. *Brain Res. Rev.*, 2001.

[7] G. Cheng, J. Han, L. Guo, and T. Liu. Learning coarse-to-fine sparselets for efficient object detection and scene classification. In *CVPR*, 2015.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] M. Figurnov, M. D. Collins, Y. Zhu, L. Zhang, J. Huang, D. P. Vetrov, and R. Salakhutdinov. Spatially adaptive computation time for residual networks. In *CVPR*, 2017.

[10] J. Fu, H. Zheng, and T. Mei. Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 2017.

[11] T. Ge, K. M. Kendrick, and J. Feng. A novel extended granger causal model approach demonstrates brain hemispheric differences during face recognition learning. *PLoS Comput Biol*, 2009.

[12] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2), 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[15] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[16] L. Kauffmann, S. Ramanoël, and C. Peyrin. The neural bases of spatial frequency processing during scene perception. *Frontiers in integrative neuroscience*, 8, 2014.

[17] R. Kawakami, Y. Shinohara, Y. Kato, H. Sugiyama, R. Shigemoto, and I. Ito. Asymmetrical allocation of nmda receptor $\varepsilon 2$ subunits in hippocampal circuitry. *Science*, 300(5621), 2003.

[18] K. M. Kendrick, Y. Zhan, H. Fischer, A. U. Nicol, X. Zhang, and J. Feng. Learning alters theta-nested gamma oscillations in inferotemporal cortex. *Nature*, 2009.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. *ICCV Workshops*, 2013.

[20] A. Krizhevsky. Learning multiple layers of features from tiny images. Masters thesis, Department of Computer Science, University of Toronto, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[22] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply supervised nets. In *AISTATS*, 2015.

[25] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[26] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, 2015.

[27] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. In *ICLR*, 2017.

[28] P. Macneilage, L. Rogers, and G. Vallortigara. Origins of the left & right brain. *Scientific American*, 2009.

[29] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3D pose estimation and sub-category recognition. In *CVPR*, 2015.

[30] Naselaris. A voxel-wise encoding model for early visual areas decodes mental images of remembered scenes. *NeuroImage*, 2015.

[31] A. Odena, D. Lawson, and C. Olah. Changing model behavior at test-time using reinforcement learning. *CoRR*, abs/1702.07780, 2017.

[32] C. Peyrin, C. M. Michel, S. Schwartz, G. Thut, M. Seghier, T. Landis, C. Marendaz, and P. Vuilleumier. The neural substrates and timing of top–down processes during coarse-to-fine categorization of visual scenes: A combined fMRI and ERP study. *Journal of cognitive neuroscience*, 22(12), 2010.

[33] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger. Memory-efficient implementation of densenets. *CoRR*, abs/1707.06990, 2017.

[34] Y. Shinohara, H. Hirase, M. Watanabe, M. Itakura, M. Takahashi, and R. Shigemoto. Left-right asymmetry of the hippocampal synapses with differential subunit allocation of glutamate receptors. *PNAS*, 105(49), 2008.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. M. A. Patwary, Prabhat, and R. P. Adams. Scalable bayesian optimization using deep neural networks. In *ICML*, 2014.

[37] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[38] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[39] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[40] A. W. Toga and P. M. Thompson. Mapping brain asymmetry. *Nature Reviews Neuroscience*, 4(1), 2003.

[41] M. Turgeon. *Right Brain/Left Brain Reflexology*. Inner Traditions/Bear & Co, 1993.

[42] A. Veit, M. J. Wilber, and S. J. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NIPS*, 2016.

[43] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[44] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang. Multiple granularity descriptors for fine-grained categorization. In *ICCV*, 2015.

[45] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015.

[46] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*, 2015.

[47] A. R. Zamir, T.-L. Wu, L. Sun, W. Shen, J. Malik, and S. Savarese. Feedback networks. In *CVPR*, 2017.

[48] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking deep filter responses for fine-grained image recognition. In *CVPR*, 2016.

[49] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.