



COMO: Efficient Deep Neural Networks Expansion With COnvolutional MaxOut

Baoxin Zhao , Haoyi Xiong , Member, IEEE, Jiang Bian, Zhishan Guo , Member, IEEE, Cheng-Zhong Xu, Fellow, IEEE, and Dejing Dou

Abstract—In this paper, we extend the classic MaxOut strategy, originally designed for Multiple Layer Preceptors (MLPs), into COnvolutional MaxOut (COMO) — a new strategy making deep convolutional neural networks wider with parameter efficiency. Compared to the existing solutions, such as ResNeXt for ResNet or Inception for VGG-alikes, COMO works well on both linear architectures and the ones with skipped connections and residual blocks. More specifically, COMO adopts a novel *split-transform-merge* paradigm that extends the layers with *spatial resolution reduction* into multiple parallel splits. For the layer with COMO, each split passes the input feature maps through a *4D convolution operator* with independent *batch normalization operators* for transformation, then merge into the aggregated output of the original sizes through *max-pooling*. Such a strategy is expected to tackle the potential classification accuracy degradation due to the spatial resolution reduction, by incorporating the multiple splits and max-pooling-based feature selection. Our experiment using a wide range of deep architectures shows that COMO can significantly improve the classification accuracy of ResNet/VGG-alike networks based on a large number of benchmark datasets. COMO further outperforms the existing solutions, e.g., Inceptions, ResNeXts, SE-ResNet, and Xception, that make networks wider, and it dominates in the comparison of accuracy versus parameter sizes.

Manuscript received October 30, 2019; revised March 18, 2020 and April 30, 2020; accepted May 27, 2020. Date of publication June 15, 2020; date of current version May 26, 2021. This work was supported in part by the National Key R&D Program of China under Grant 2019YFB2102100, in part by the National Key R&D Program of China under Grant 2018YFB1402600, in part by the Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence, and in part by the Shenzhen Engineering Research Center for Beidou Positioning Service Improvement Technology under Grant XMHT20190101035. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lamberto Ballan. (Baoxin Zhao and Haoyi Xiong contributed equally to this work.) (Corresponding author: Haoyi Xiong.)

Baoxin Zhao was with the Big Data Lab, Baidu Inc., Beijing 100085, China now with the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: bx.zhao@siat.ac.cn).

Haoyi Xiong and Dejing Dou are with the Big Data Lab, Baidu Inc., Beijing 100085, China, and also with the National Engineering Lab of Deep Learning Technology and Applications, Beijing, China (e-mail: xhyccc@gmail.com; doudejing@baidu.com).

Jiang Bian was with the Big Data Lab, Baidu Inc., Beijing 100085, China, and now with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mail: bjbj1111@knights.ucf.edu).

Zhishan Guo is with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mail: zsguo@ucf.edu).

Cheng-Zhong Xu is with the State Key Lab of IOTSC, Faculty of Science and Technology, University of Macau, Macau SAR 999078, China (e-mail: czxu@um.edu.mo).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMM.2020.3002614

Index Terms—Artificial neural networks, computational and artificial intelligence, feedforward neural networks, multilayer perceptrons, neural networks.

I. INTRODUCTION

DEEP convolutional neural networks (Deep CNNs) [1] have been widely examined as the major workhorses for image classification and pattern recognition. From the simple handwritten digits recognition task [2] to the ILSVRC challenges [3], the architectures of convolutional neural networks have evolved to adapt the increasing complexity of the datasets. While architectures, such as LeNet [2], AlexNet [4], ResNet [5], and DenseNet [6], with novel components have been invented, researchers also made significant efforts to design new strategies to extend the existing architectures for higher capacity. In this work, we intend to study a practical paradigm to widen the CNNs for higher accuracy.

As early as 2013, the maxout strategy has been proposed by [7] as a potential replacement for the rectified linear units (ReLUs) used in common Multi-Layer Preceptors (MLPs), under the dropout settings, so as to enhance the capacity of MLPs. More specifically, within an MLP, maxout *splits* the activation (of a) layer into multiple independent paths, where each path is with a weight matrix for multiplication. Then maxout *merges* the results received from multiple paths through *max-pooling* and forwards to the next layer. In this work, we term the above practice as a “*split-transform-merge*” strategy. Such strategy *defacto* widens the affected layers of MLP, while preserving the rest of the architecture.

While such a strategy improves the capacities of deep MLPs through widening some bottlenecked layers, Maxout does not work on convolutional neural networks straightforwardly, as the weight matrix multiplication might not be a suitable operator for CNNs. To widen the deep CNNs with the “*split-transform-merge*” alike paradigms, for example, Inception [8] has been proposed to enhance the common linear architectures, such as VGG-alikes [9]. More specifically, Inception splits the vanilla convolutional layer used by the linear convolutional architectures into multiple paths with independent convolutional operators, then merges the results from these paths via *concatenation*, and forwards the merged result to the next layer. On the other hand, ResNeXt [10] has been proposed to improve ResNet [5] through aggregating multiple convolutional paths in a similar way. While these extensions significantly improved the accuracy of deep CNN over some benchmark datasets [8], [10] and

have inspired new architectures and algorithms [11], [12], the empirical design patterns behind such improvement have not yet been deeply studied or exactly analyzed. In our research, we intend to study novel “split-transform-merge” strategies from the following aspects:

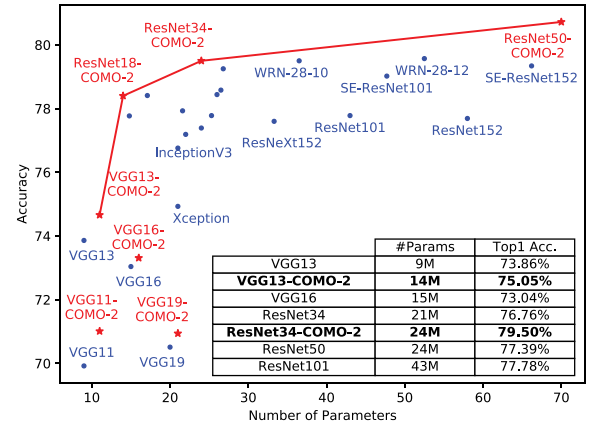
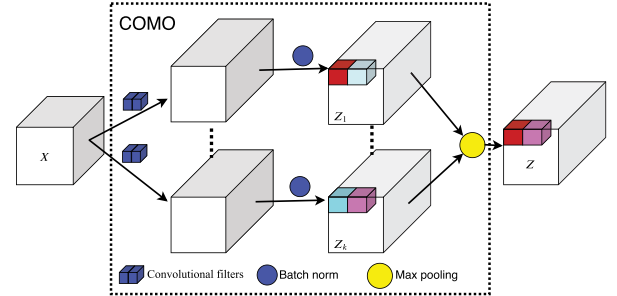
- **Architectures:** While Inception and ResNeXt have been considered as the extension of some specific architectures such as common linear architectures (e.g., VGG) and ResNet, we are wondering whether there exists any method that works well both on linear architectures and residual learning-based ones.
- **Layers:** Instead of widening some key layers in the neural networks, Inception and ResNeXt almost extend every meaningful layer in the architectures (excepting the layer for pre-processing and FC layer for classification). Such practice frequently makes the models extremely over-parameterized. Our research is wondering: (a) whether we can improve the performance by enhancing some of the layers; and (b) how to identify the layers for modification.
- **Interpretability:** Any good practice deserves good interpretability. Apparently, there needs to be some direct evidence to demonstrate the reasons why the network with split-transform-merge would outperform the original one. We plan to compare the feature maps and activation of the original networks to the enhanced one. We expect to see what exactly our proposed split-transform-merge could improve and how it can finally lead to better classification accuracy.

Our Contribution: In this work, we propose CONvolutional MaxOut (COMO) to boost the classification accuracy for deep CNNs through widening some key layers. The key contributions are made as follows.

- **Generic Architectures and Specific Layers:** COMO works well on both VGG and ResNet-alikes, where it only extends the layers with *spatial resolution reduction*: the size of each feature map has been reduced from inputs through the outputs. Compared to Inception and ResNeXt, COMO widens and improves the capacity of CNNs while incorporating fewer parameters. Our latter ablation study shows, with COMO, widening only the spatial resolution reduction layers performs significantly better than a model that extends all the layers.
- **Simple yet Effective Transform and Merge Operators:** To enable the classic maxout strategy onto the deep CNNs, COMO incorporates a new set of operators. COMO first branches a convolutional layer (convolution operators+activation) into N independent paths ($N \geq 2$) and assigns each path a convolutional operator. Furthermore, COMO continues each path with batch normalization operators and merges the results received from paths into the feature maps of original sizes through a max-pooling operator. Please refer to Table I for the comparison in details from operators’ perspectives.
- **Advantage and Interpretability:** With COMO, one can easily extend deep CNNs of VGG and ResNet-alikes with significant classification accuracy boosting. VGG and ResNet with COMO can even outperform the existing extent models such as Inception and ResNeXt. Fig. 1(b)

TABLE I
OPERATORS USED BY SPLIT-TRANSFORM-MERGE STRATEGIES: MAXOUT VS. INCEPTION VS. RESNEXT VS. COMO: PATH BN (BATCH NORMALIZATION) THAT INDICATES WHETHER A BATCH NORMALIZATION OPERATOR IS ASSIGNED TO EACH PATH; Mat_M REFERS TO THE WEIGHT MATRIX MULTIPLICATION OPERATOR; Conv REFERS TO THE CONVOLUTIONAL OPERATORS OF VARIOUS SIZES; MP REFERS TO THE MAX-POOLING OPERATOR; Concat REFERS TO CONCATENATION

	Transform	Path BN.	Merge
Maxout	Mat_M	No	MP
Inception	MP*+Conv	Yes	Concat
ResNeXt	Conv	Yes	Concat+Conv*
COMO	Conv	Yes	MP



(b) Testing Accuracy versus Parameter Sizes on CIFAR-100

Fig. 1. Design of COMO with Two Splits and Performance Comparison on CIFAR-100 Datasets: VGG13-COMO-2 is read as the VGG13 network, enhanced using COMO with 2 independent paths; ResNet18-COMO-2 is read as the ResNet18 network extended using COMO with 2 independent paths; WRN-28-10 is read as Wide ResNet with 28 layers and width 10; all blue points stand for the results based on the existing models such as VGG, ResNet, Wide ResNet, Inception, ResNeXt and etc.; all models are trained with 200 epochs from scratch without using data augmentation or other tricks.

clearly illustrates the comparison of testing accuracy over parameter sizes for a wide range of well-trained models based on CIFAR-100 datasets, where COMO networks dominate in the comparison. In addition to an advantage in classification accuracy, COMO also generates good interpretability [13]. Through dissecting COMO with activation maps drawn from the networks, one can find that COMO networks can detect significantly more “visual concepts” than the vanilla VGG and ResNet. Please refer to the experiment section for details.

To summarize, our contributions include not only the design of COMO, but also the examined accuracy enhancement due to the use of COMO and the improvement it provides in interpretability. Note that the comparison to automated deep learning solutions [12], [14] are out of scope, as this work focuses on the empirical design and practical paradigms to enhance the capacities of CNNs under “*split-transform-merge*” settings.

II. PRELIMINARIES AND RELATED WORK

In this section, we review the recent studies that are relevant to our work, where we intend to summarize the architectural efforts made to improve the capacity of existing networks.

As was mentioned, maxout [7], Inception [8], and ResNeXt [10] are the *split-transform-merge* strategies that widen the architectures of MLP, common linear convolutional architectures (e.g., VGG [9]), and the ResNet [5] respectively. Though all these strategies follow the “split-transform-merge” patterns, the operators used are, indeed, quite different. Table I lists the operators used. Compared to Inception and ResNet which were designed for deep CNNs, maxout targets at approximation the arbitrary activation for MLPs. Thus, the vector-matrix multiplication is used as the transform, while maxout simply merges the results from multiplication through max-pooling. It has been observed that making deep neural networks wide can enhance the capacity of the networks to fit the training data [15] while incorporating redundant parameters and often poor generalization.

For deep CNNs, Inception and ResNeXt are two basic models that extend linear convolutional architectures and ResNets. After splitting the input from the previous layer into multiple parallel paths, Inception and ResNeXt both use depthwise separable convolution operators to transform the input. Furthermore, Inception also incorporates an additional max-pooling operator before the convolution operators in one of the paths [8]. To balance these paths, Inception and ResNeXt also adopt a path-wise batch normalization to normalize the results of each path. Later both Inception and ResNeXt use concatenation to merge the paths, while ResNeXt specifically adopts an additional 1×1 convolutional operator to reshape the output forwarded to the next layer (without architectural changes) [10]. In terms of methodologies, [16], [17] also proposed to use max-out with MLPs to improve CNNs for capacity enhancement or multi-modal fusion purposes respectively, while COMO incorporates a convolutional analog of maxout with convolution filters and pixel-wise max-pooling operators. Authors in [18] also proposed to use maxout strategies on the fully-connected layers to improve a hybrid architecture of CNNs and Long-Short Term Memory (LSTM) networks for lip reading. We found that the design of [16], [17] is incompatible with the model CNN architectures due to the use of weight matrix multiplication operators, which are parameter-inefficient compared to convolutional ones.

In addition to above two, InceptionResNet [19], Xception [11], and SENets [20] also follow the split-transform-merge patterns to enhance the capacity through multi-pathing the deep CNNs. Note that Wide ResNets (WRNs) [21] and EfficientNet [12] are not discussed here, as these two techniques widen the networks by incorporating more channels, rather than by

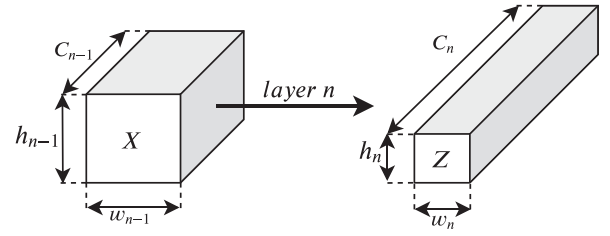


Fig. 2. A Spatial Reduction Layer: for the n th layer, w_{n-1} , h_{n-1} , and C_{n-1} refer to the width, height, and the number of channels of the input tensor (X), while w_n , h_n , C_n are for the output tensor (Z).

splitting them into multiple paths. Furthermore, our work focuses on the empirical design patterns and practical paradigms to widen the CNNs, such as the extension of Xception and ResNeXt to linear CNN and ResNet design. Thus, the automated deep learning tools such as EfficientNet [12] and GPipe [14] (which usually perform better than empirical designs) are not included in the comparison here.

Compared to our work, the most relevant studies are indeed Inception and ResNeXt [8], [10]. As was discussed in Table I, the operators used by COMO are indeed simpler than these two. More specifically, in each splitting path, COMO only uses one convolution operator, while Inception and ResNeXt adopt the depth-wise separable ones which usually incorporate a *down-sampling and up-sampling* procedure and might cause potential accuracy loss. Furthermore, COMO uses max-pooling as the merge operator, while the other two need to use concatenation or even concatenation with 1 convolution to reshape the outputs. To the end, as was illustrated in Fig. 1(b), COMO enjoys higher classification accuracy improvement with parameter efficiency.

III. COMO: DESIGN AND MECHANISM

In this section, we first introduce the definition of spatial resolution reduction layers with examples in VGG and ResNets. Then, we present the design of COMO operators, and how COMO would be incorporated with existing VGG and ResNet networks.

A. Spatial Resolution Reduction Layer

We identify whether a layer is using Spatial Resolution Reduction on its input and output tensors. As shown in Fig. 2, given a layer with spatial resolution reduction (e.g., the n th layer), the spatial resolution of the input tensor $w_{n-1} \times h_{n-1}$ has been reduced to the low resolution $w_n \times h_n$ in the output. On the other hand, the number of channels would increase from the input C_{n-1} to the output C_n .

Fig. 3 demonstrates the examples of spatial resolution reduction layers in two deep CNNs, where we can see the spatial resolution reduction layers highlighted in dash boxes. Both VGG13 and ResNet18 consist of three spatial resolution reduction layers. While VGG13 incorporates spatial resolution reduction in the 3rd, the 5th and the 7th layers respectively, ResNet18 reduces its spatial resolution in the layers associated to the 3rd, the 5th and the 7th skipped connections.

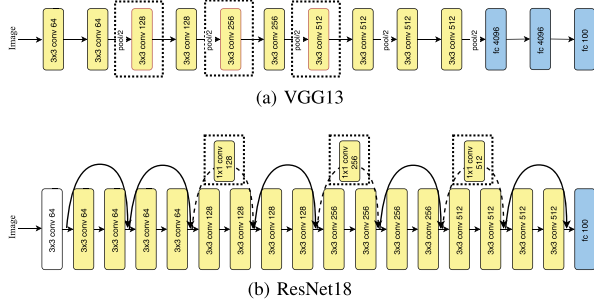


Fig. 3. Example of Spatial Resolution Reduction Layers (in the dash boxes).

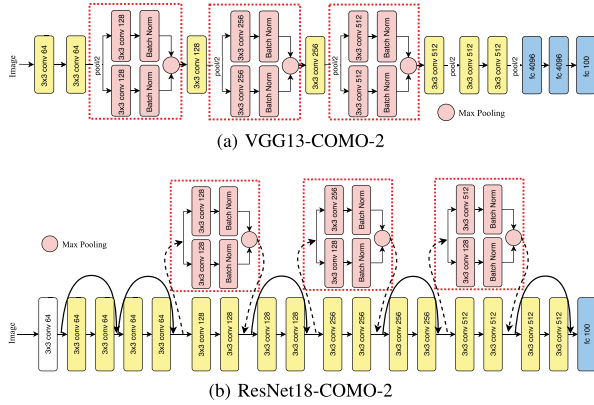


Fig. 4. Example of VGG and ResNet with COMO.

Please note that, by definition, spatial resolution reduction layers are NOT necessary to be pooling layers, and vice-versa. For ResNet, COMO only refactors the spatial-resolution layers in the spliced connections which originally use convolutional filters with stride = 2 to reduce the spatial resolution. VGG networks also use a max-pooling operator with stride = 2 to reduce the size of feature maps. In both examples above, the spatial resolution reduction layers are not the pooling layers.

B. COMO Transform and Merge Operators

As was mentioned, COMO incorporates a set of simple yet effective operators for split-transform-merge paradigms. More specific, Fig. 1(a) illustrates an example of a layer extended by COMO using two independent splits. Each split first transforms the input tensor using a 3×3 convolution operator of 4D tensor, then normalizes the results through a batch normalization operator. The two splits are finally merged into the results through an element-wise max-pooling operator.

Fig. 4 illustrates the example of VGG13-COMO-2 and ResNet18-COMO-2 respectively. It has been shown that the spatial resolution reduction layers are all replaced by the COMO operators. With additional splits and operators, VGG13-COMO-2 uses 2 M more parameters while ResNet18-COMO-2 uses 3 M more parameters. Similar extension could be made on VGG, ResNet and other architectures to boost the capacity of networks with certain parameter efficiency.

Please note that, though COMO uses high-resolution feature maps and more learnable parameters, it only incorporates a very small number of additional FLOPs for computation. Indeed,

TABLE II
STATISTICS ON SOURCE/TARGET DATASETS

Datasets	# Objects	# Train/Test
CIFAR-100	100	50K/10K
Food-101	101	75K+/25K+
Stanford Dogs-120	120	12K/8.5K
Caltech-60	256	30K+
Caltech-30	256	30K+
MIT Indoors	67	5K+/1K+
CUB-200-2011	200	11K+

COMO improves the spatial-resolution layers through preserving the resolution of feature maps, and it uses the pixel-level max-pooling over feature maps to finally merge the split paths. Compared to other split-transform-merge design, COMO uses max-pooling rather than concatenation to reduce the size of immediate results, and to lower the computational complexity relatively.

IV. EXPERIMENTS

In this section, we present the experiment and comparison results. We first present the experiment settings with baseline algorithms and datasets introduced, then demonstrate the overall accuracy of COMO with parameter sizes compared, finally we conduct ablation studies to show the functionalities of each key component.

A. Experiment Setups

In this study, we compared COMO networks with the baseline algorithms as follow: VGG networks including VGG11, VGG13, VGG16, and VGG19; ResNet networks including ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152; Inception, including InceptionV3, and InceptionV4; ResNeXt, including ResNeXt50, ResNeXt101, and ResNeXt152; Xception; as well as SE-ResNet, including SE-ResNet34, SE-ResNet50, and SE-ResNet152.

To evaluate the performance improvement made by COMO, we extend part of the above VGG and ResNet networks with the COMO extensions using two independent splits, i.e., VGG11-COMO-2, VGG13-COMO-2, VGG16-COMO-2, VGG19-COMO-2, ResNet18-COMO-2, ResNet34-COMO-2, ResNet50-COMO-2, and ResNet101-COMO-2 networks. All networks are implemented in the way aforementioned. We did not include the experiments based on DenseNet [6] here, as DenseNet is already over-parameterized with redundant paths/splits.

All the above networks are evaluated and compared using the following datasets:

- *CIFAR-100*: The CIFAR-100 dataset consists of 60,000 32×32 color images in 100 classes, with 600 images per class. There are 500 training images and 100 testing images per class [22].
- *Caltech-256*: Caltech 256 is a dataset with 256 object categories containing a total of 30,607 images. Different numbers of training examples are used by researchers to validate the generalization of proposed algorithms. In this paper, we create two configurations for Caltech 256, which

TABLE III
TESTING ACCURACY (%) ON CIFAR-100

Model	Params	GFLOPs	CIFAR-100
VGG11	9M	0.31	69.92
VGG13	9M	0.46	73.86
VGG11-COMO-2	11M	0.42	71.01
VGG13-COMO-2	11M	0.57	74.66
ResNet18	11M	1.11	75.61
ResNet18-COMO-2	14M	1.32	78.40
ResNeXt50	15M	7.37	77.77
VGG16	15M	0.63	73.04
VGG16-COMO-2	16M	0.74	73.31
VGG19	20M	0.80	70.51
VGG19-COMO-2	21M	0.91	70.94
Xception	21M	20.53	74.97
ResNet34	21M	2.32	76.76
SE-ResNet34	22M	2.32	77.93
InceptionV3	22M	6.78	77.19
ResNet34-COMO-2	24M	2.53	79.50*
ResNet50	24M	2.60	77.39
ResNeXt101	25M	14.78	77.78
SE-ResNet50	27M	2.60	78.58
ResNeXt152	33M	20.46	77.60
InceptionV4	41M	15.00	75.86
ResNet101	43M	5.02	77.78
SE-ResNet101	48M	5.02	79.02
ResNet152	58M	7.44	77.69
SE-ResNet152	66M	7.44	79.34
ResNet50-COMO-2	70M	6.59	80.72**
ResNet101-COMO-2	89M	9.01	80.51

have 30 and 60 random sampled training examples respectively for each category, following the procedure used in [23].

- *Stanford Dogs-120*: The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. There are exactly 100 examples per category for training [24].
- *MIT Indoors-67*: MIT Indoors 67 is a scene classification task containing 67 indoor scene categories, each of which consists of 80 images for training and 20 for testing [25].
- *Caltech-UCSD-Birds-200-2011*: CUB-200-2011 contains 11,788 images of 200 bird species. Each species is associated with a Wikipedia article and organized by scientific classification [26]. Each image is annotated with a bounding box, part location, and attribute labels. We use only classification labels during training.
- *Food-101*: Food-101 is a large scale data set of 101 food categories, with 101,000 images, 750 training images and 250 test images are provided for each class [27].

All networks are trained using fine-tuned hyper-parameters. A portion of the experimental results is cross-verified with github.¹ All networks have been trained using 200 epochs under the same settings with both PaddlePaddle and Pytorch implementations. Learning rate decay has been triggered twice at the 80th and the 160th epoch respectively.

B. Overall Comparisons

Table III presents the testing accuracy of the above algorithms on CIFAR-100 datasets. We conclude the testing accuracy comparison as follow.

- *COMO vs Vanilla Networks*: Using COMO with 2 splits, VGG11, VGG13, VGG16, VGG19, ResNet18, ResNet34, ResNet50, and ResNet101 all have been

¹<https://github.com/weiaicunzai/pytorch-cifar100>

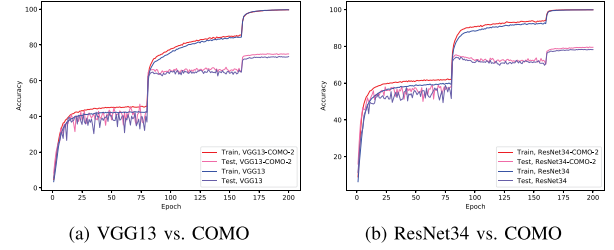


Fig. 5. The Comparison between Training and Testing Accuracy Curves on CIFAR-100.

improved with 0.3% to 3% higher testing accuracy. It has also shown that ResNet50-COMO-2 outperforms all other networks, with the best testing accuracy in the comparison, though ResNet50-COMO-2 needs slightly more parameters than other networks. For example, ResNet50-COMO-2 achieves around 1.42% higher accuracy than SE-ResNet152 and 3.03% higher accuracy than ResNet152, while consuming 4 M and 12 M more parameters respectively.

- *Parameter Efficiency*: The COMO networks outperform the networks with similar or even larger parameter sizes. For example, ResNet18-COMO-2 with 14 M parameters achieves 78.40% testing accuracy, and it outperforms ResNeXt50 (15 M, 77.77%) and VGG16 (15 M, 73.04%). ResNet34-COMO-2 with 24 M parameters achieves 79.5% testing accuracy and performs better than all baselines.
- *Computational Efficiency*: The COMO networks only incorporate a very small number of additional FLOPs to enhance the original networks, while in terms of testing accuracy they all outperform the networks with similar or even larger FLOPs. Note that the FLOPs here are measured using a mock input of size $32 \times 32 \times 3$ and estimated accordingly [28]. The overall comparison might suggest that COMO networks perform better than the handcrafted networks under limited FLOPS budgets.

We further compare COMO with baseline networks using other datasets. Table IV lists the testing accuracy comparison between VGG/ResNet and COMO networks. Similar observations have been also obtained with consistent conclusions.

C. Case Studies

To understand the performance of COMO, we conduct the following case studies.

1) *Fitting Capacity vs. Generalization Performance*: Fig. 5 demonstrates the comparison of training and testing accuracy curves between VGG13, ResNet34, and their COMO extensions. It can be clearly observed that COMO provides deep CNNs both (1) enhanced capacity to fit training data as well as (2) the generalizability to adapt testing samples.

More specifically, before the 160th epoch (i.e., the second learning rate decay), VGG13-COMO-2 incorporates a higher training accuracy than vanilla VGG13. Similar observations can be also made in the comparison between ResNet34-COMO-2 and ResNet34. It indicates that COMO networks fit the training datasets with a closer gap. Furthermore, after the 160th epoch,

TABLE IV
TESTING ACCURACY (%) ON OTHER DATASETS

VGG	Params.	Food-101	Stanford Dogs	Caltech60	Caltech30	Indoors	CUB-200-2011
VGG11	9M	80.17	52.98	57.26	40.16	48.69	46.77
VGG13	9M	81.90	54.84	57.28	39.95	50.79	44.30
VGG11-COMO-2	11M	80.77	55.63	57.35	40.18	50.56	47.72*
VGG13-COMO-2	11M	82.92*	56.69*	57.99*	41.25*	51.16*	46.39
VGG16	15M	82.04	51.92	55.49	37.09	46.15	40.93
VGG16-COMO-2	16M	82.90	52.19	56.08	38.37	47.67	41.42
VGG19	20M	81.84	47.62	52.97	32.30	42.56	38.05
VGG19-COMO-2	21M	82.19	46.79	53.19	37.77	43.75	36.93
ResNet	Params.	Food-101	Stanford Dogs	Caltech60	Caltech30	Indoors	CUB-200-2011
ResNet18	11M	82.81	59.42	62.11	42.65	57.29	51.42
ResNet18-COMO-2	14M	84.03	60.72	64.05	42.87	60.28	51.76
ResNet34	21M	84.09	58.73	63.21	44.87	57.44	49.03
ResNet34-COMO-2	24M	85.08	62.83	64.36	45.46*	60.28	51.83*
ResNet50	24M	85.42	61.05	63.47	42.48	59.61	46.27
ResNet101	43M	85.67	62.60	64.64	44.50	58.94	46.46
ResNet50-COMO-2	70M	86.39	62.20	64.31	43.76	59.76	48.43
ResNet101-COMO-2	89M	86.49*	63.07*	67.07*	44.66	61.63*	47.36

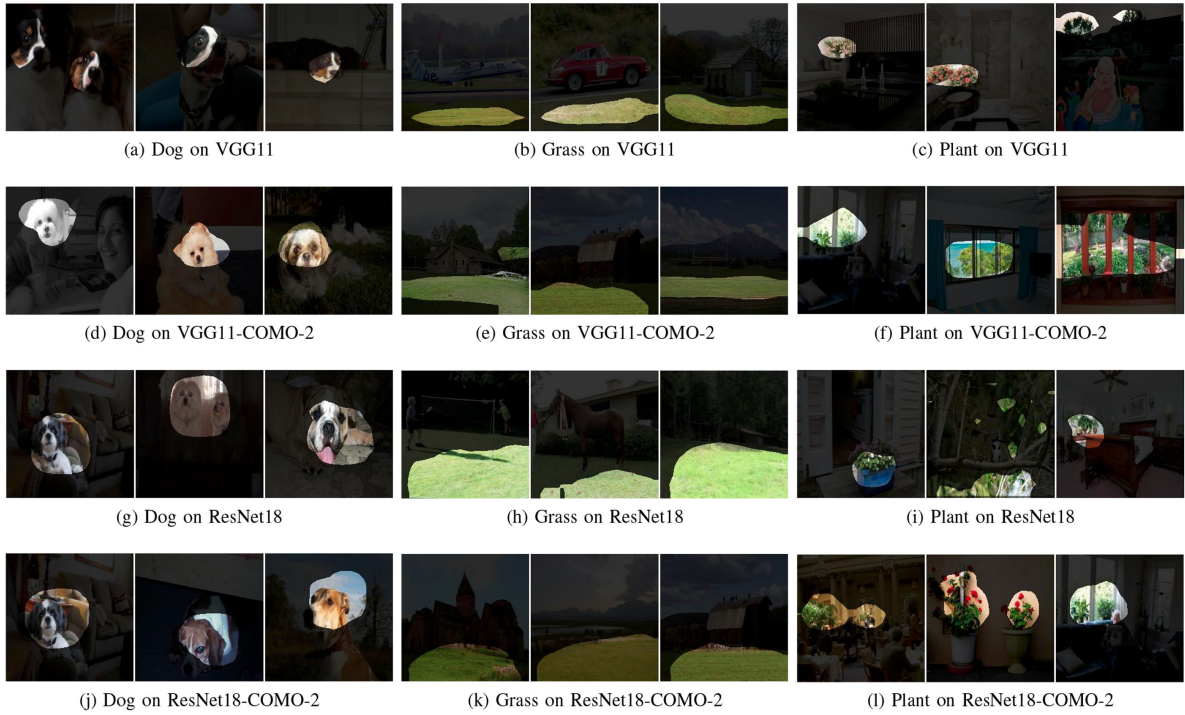


Fig. 6. Examples of Activation Maps. For the comparison of activation maps, we used the standard implementation of Network Dissection [13] provided at <http://netdissect.csail.mit.edu/>. For every architecture in the experiments, the Network Dissection tool outputted a ranking list of images categorized by the tested visual concepts, while the rank is based on the Intersection over Union (IoU) of the images' activation maps using the architecture. Only top-ranked images are visualized. For the overall detection capacity of the visual concepts, please refer to Fig. 7 for the comparison.

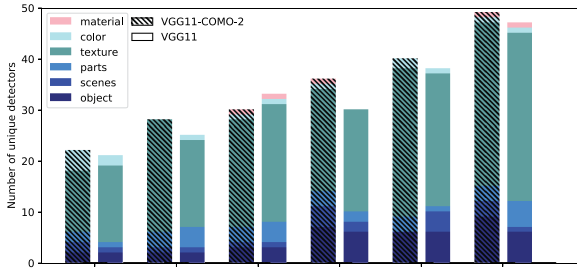
the generalization accuracy of COMO networks is significantly higher than the vanilla VGG/ResNet. It thus indicates better generalization performance.

2) *Network Dissection and Concept Detection*: To further understand the performance of COMO and how it works, we use network dissection tools [13] to analyze the feature maps drawn from COMO networks.

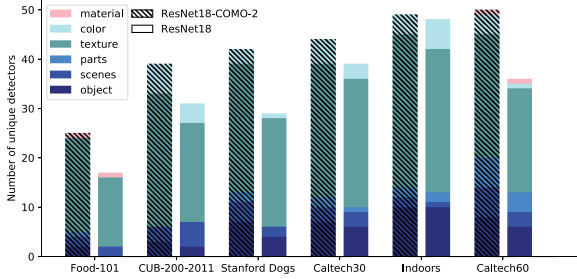
We first run the inference procedure of VGG11, ResNet18 networks and their COMO extensions using a set of object detection images provided in [13], where these networks have been well-trained using the aforementioned datasets. Then, we collect the activation maps from the last convolutional layer right before the fully connected layers. According to [13], these

activation maps have been considered as the criterion for the decision-making of classification. Following the settings of [13], we threshold the activation in the activation maps and trace back the visual pixels that are “selected” for classification. In this way, we can visualize “to where” the networks pay attention, toward the classification of every image. Then, we could follow-up the objects of different types that have been covered by these pixels, then identify the visual concepts, such as material, color, etc., that have been learned by the networks.

Fig. 6 demonstrates the example of activation maps, where the pixels to which the networks pay attention have been highlighted. These examples are randomly selected from the massive testing images and categorized by the content objects.



(a) VGG11 vs VGG11-COMO-2



(b) ResNet18 vs ResNet18-COMO-2

Fig. 7. Visual concepts detected.

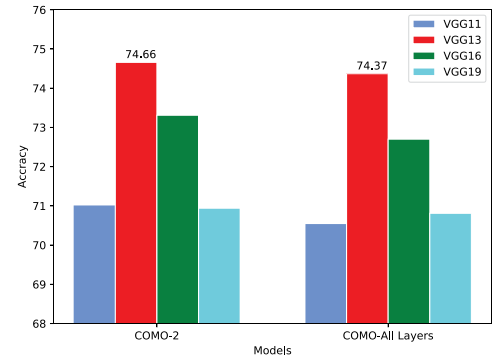
Compared COMO networks with the vanilla ones, COMO enlarges the area of activation and covers a great portion of visual objects (especially for comparison between VGG11 and VGG11-COMO-2). Based on the activation maps, we intend to specify the visual concepts detected by the networks using the activation maps. Fig. 7 illustrates the number of different visual concepts that have been detected through the trained networks. It has been observed that in general the COMO networks detect more visual concepts than the vanilla one, for all training datasets. We believe that with more visual concepts being detected, the COMO networks are capable of classifying images with higher accuracy.

Remark: Until now, we have analyzed the performance advantages of COMO networks from both learning and inference perspectives, where we can see the generalization and fitting capacity advantage of COMO, as well as its ability to detect visual concepts, on top of wide training datasets.

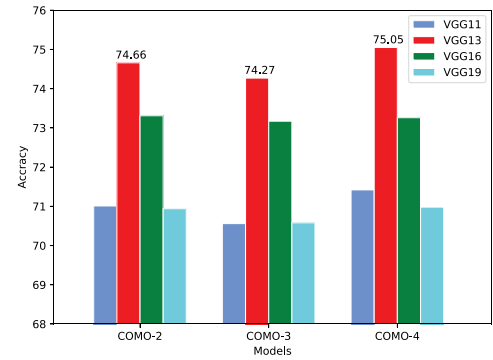
D. Ablation Studies

To understand some open issues of COMO, we conduct ablation studies as follow.

1) *Affected Layers:* COMO targets at improving the spatial resolution reduction layers, while leaving the rest of the architecture unmodified. We assume the spatial resolution reduction might cause potential accuracy loss, while the use of COMO would enhance the capacity of the network in such layers. Our ablation study evaluates the performance of using COMO to extend all layers in VGG11, VGG13, VGG16, and VGG19, then compares them with the standard COMO using two splits. Fig. 8(a) shows that extending spatial resolution reduction layers with COMO (denoted as COMO-2) outperforms the ones that extend every layer with COMO (denoted as COMO-All-Layers). Such observation further validates our intuition and approach.



(a) On Affected Layers



(b) On the Number of Splits

Fig. 8. Ablation studies.

TABLE V
TESTING ACCURACY (%) ON CIFAR-100 FOR COMO WITH AND WITHOUT BATCH NORMALIZATION (BN)

	with BN	w/o BN
VGG11-COMO-2	71.01	69.49
VGG13-COMO-2	74.66	73.73
VGG16-COMO-2	73.31	72.39
VGG19-COMO-2	70.94	69.93

2) *Number of Splits:* In the above examples, we present the networks using COMO with two independent splits. COMO can further scale-up with more splits while enjoying better capacity enhancement. Our ablation study evaluates COMO networks with 2, 3 and 4 independent splits. The comparison result shown in Fig. 8(b) demonstrates that COMO networks with 3 splits usually perform worse than the ones with 2 splits, while COMO networks with 4 splits would perform significantly better the other two. In general, we could expect higher testing accuracy when using more splits in COMO networks. However, the number of parameters would be linearly increased with the number of splits. In practice, we don't recommend adopting COMO with a large number of splits unless the parameter size is not an issue.

3) *Batch Normalization and Mixup:* In our study, we also evaluate COMO with batch normalization disabled. The testing accuracy significantly decreases in such settings (shown in Table V), since the max-pooling operator would fail to pick up informative features for classification without the batch normalized split. As many deep architectures, our studies show that a common mixup strategy could help to augment the training datasets for COMO networks and further improve the testing accuracy. Furthermore, as shown in Table VI, COMO

TABLE VI
TESTING ACCURACY (%) ON CIFAR-100 WITH MIXUP

VGG	Params	CIFAR-100
VGG11	9M	71.70
VGG13	9M	74.80
VGG11-COMO-2	11M	72.81
VGG13-COMO-2	11M	75.76
ResNet18	11M	78.88
ResNet18-COMO-2	14M	79.40
VGG16	15M	73.40
VGG16-COMO-2	16M	74.02
VGG19	20M	69.83
VGG19-COMO-2	21M	70.90
ResNet34	21M	79.96
ResNet34-COMO-2	24M	80.66
ResNet50	24M	79.67
ResNeXt101	25M	80.00
ResNet50-COMO-2	70M	81.79
ResNet101-COMO-2	89M	81.97

still outperforms the vanilla networks in the testing accuracy comparison from both accuracy and parameter efficiency perspectives.

V. DISCUSSION AND CONCLUSION

In this paper, we propose a novel maxout strategy, namely COMO (COvolutional MaxOut), which enhances the capacity of deep CNNs with parameter efficiency improved. As the operators of vanilla COMO are usually incompatible with deep CNNs, COMO introduces a simple yet effective set of *split-transform-merge* operators, widening the layers with spatial resolution reduction. Extended with COMO, the spatial resolution reduction layers are converted into multiple independent splits. Each split first feed-forwards the input feature map to a 4D convolution operator, then transforms the result through a batch normalization operator, and finally merges into the aggregated output through max-pooling.

The design goal is to tackle the possible degradation of the classification accuracy caused by the spatial resolution reduction, as it enhances the capacity of feature learning with the multiple splits. Moreover, the superiority is shown compared with the existing ResNeXt for ResNet or Inception for VGG-alikes, where COMO is suitable for both linear architectures and the architectures with residual connections. To demonstrate the real effect of COMO in a variety of deep CNNs, we conduct a series of intensive experiments on the commonly used benchmark datasets. The result shows that COMO can remarkably increase the classification accuracy of ResNet/VGG-alike networks, and that our model also dominate in comparison with the well-known solutions (ResNeXts and Inceptions). The additional ablation study offers tremendous insights into why and how COMO works. All empirical results and observations back up our intuition and design principles. Note that this work focuses on studying empirical design and practical paradigms followed by split-transform-merge patterns to widen the CNNs for capacity enhancements, and we further test the proposed COMO design on the image classification benchmark datasets of moderate sizes. In our future work, we hope to utilize COMO with automated machine learning algorithms, such as EfficientNet [12] and GPipe [14], on larger datasets [4], [29].

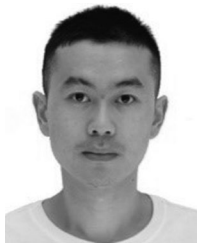
ACKNOWLEDGMENT

Dr. Baoxin Zhao was a Research Intern at the Big Data Lab at Baidu Research when the work was partly performed. Please check out PaddleClas <https://github.com/PaddlePaddle/PaddleClas> for the latest updates on the codes and models.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [3] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 248–255.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
- [6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2261–2269.
- [7] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, 2013, vol. 28, pp. 1319–1327.
- [8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2818–2826.
- [9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [10] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1492–1500.
- [11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1251–1258.
- [12] M. Tan and Quoc V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.
- [13] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. Comput. Vision Pattern Recognit.*, 2017, pp. 3319–3327.
- [14] Y. Huang *et al.*, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 103–112.
- [15] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.
- [16] W. Sun, F. Su, and L. Wang, "Improving deep neural networks with multilayer maxout networks," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, 2014, pp. 334–337.
- [17] V. Vielzeuf, A. Lechervy, S. Pateux, and F. Jurie, "CentralNet: A multi-layer approach for multimodal fusion," in *Proc. Eur. Conf. Comput. Vision*, 2018.
- [18] I. Fung and B. Mak, "End-to-end low-resource lip-reading with maxout CNN and LSTM," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2511–2515.
- [19] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.
- [20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7132–7141.
- [21] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vision Conf.*, 2016, pp. 2825–2834.
- [22] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Tech. Rep., 2009.
- [23] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 413–420.
- [24] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. 1st Workshop Fine-Grained Vis. Categorization, IEEE Conf. Comput. Vision Pattern Recognit.*, Tech. Rep. Caltech-UCSD Birds-200-2011, Colorado Springs, CO, USA, Jun. 2011.

- [25] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 413–420.
- [26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," California Institute Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
- [27] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101—Mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 446–461.
- [28] M. Thoma, "Analysis and optimization of convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2113–2122.
- [29] W. Li *et al.*, "Webvision challenge: Visual learning and understanding with web data," 2017, *arXiv:1705.05640*.



Baoxin Zhao received the B.S. degree in electric science and technology from Shandong Agriculture University, Shandong Province, in 2011, the M.S. degree in computer science and technology from the Guangdong University of Technology, Guangdong Province, in 2014, the Ph.D. degree from the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Guangdong Province, in 2020. His research interests include cloud computing, machine learning, artificial intelligence and trajectory mining.



Haoyi Xiong (Member, IEEE) received the Ph.D. degree in computer science from Telecom SudParis, University of Paris VI, Paris, France, in 2015. From 2016 to 2018, he was an Assistant Professor with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA (formerly known as University of Missouri at Rolla). From 2015 to 2016, he was a Post-Doctoral Research Associate with the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. He is currently a Principal R&D

Architect and Researcher with Big Data Laboratory, Baidu Research, Beijing, China. His current research interests include automated deep learning (AutoDL), ubiquitous computing, artificial intelligence, and cloud computing. He has published more than 60 papers in top computer science conferences and journals, such as ICML, ICLR, UbiComp, RTSS, AAAI, IJCAI, ICDM, PerCom, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY, ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATABASES, and etc. He gave keynote speeches in a series of academic and industrial activities, such as the industrial session of the 19th IEEE International Conference on Data Mining (ICDM'19), and served as Poster Co-Chair for the 2019 IEEE International Conference on Big Data (IEEE Big Data'19). Dr. Xiong was a recipient of the Best Paper Award from IEEE UIC 2012, the Outstanding Ph.D. Thesis Runner Up Award from CNRS SAMOVAR 2015, and the Best Service Award from IEEE UIC 2017. He was the Co-Recipient of Science & Technology Advancement Award (First Prize) from the Chinese Institute of Electronics 2019.



Jiang Bian received the B.Eng degree in logistics systems engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2014, and the M.Sc degree in industrial systems engineering from the University of Florida at Gainesville, FL, USA, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer and Electrical Engineering, University of Central Florida, Orlando, FL, USA, under co-supervision of Dr. Zhishan Guo and Dr. Haoyi Xiong. From 2016 to 2018, he spent the first two years of his Ph.D study in the Department of

Computer Science, Missouri University of Science and Technology, Rolla, MO, USA. His research interests include Human-Subject Data Learning, Ubiquitous Computing, Intelligent Cyber-Physical Systems.



2018, he was an Assistant Professor with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA (formerly known as University of Missouri at Rolla). His current research interests include real-time and cyber-physical systems, neural networks, and computational intelligence.

Zhishan Guo (Member, IEEE) received the B.Eng. degree in computer science and technology from Tsinghua University, Beijing, China, in 2009, the M.Phil. degree in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2011, and the Ph.D. degree in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2016. He is currently an Assistant Professor with the Department of Computer and Electrical Engineering, University of Central Florida, Orlando, FL, USA. From 2016 to



Cheng-Zhong Xu (Fellow, IEEE) received the PhD degree from the University of Hong Kong, in 1993. He is the Dean of Faculty of Science and Technology and the Interim Director of Institute of Collaborative Innovation, University of Macau, and a Chair Professor of Computer and Information Science. He was a professor of Wayne State University and the Director of Institute of Advanced Computing of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences before he joined UM in 2019. Dr. Xu is a Chief Scientist of Key Project on Smart City of

MOST, China and a Principal Investigator of the Key Project on Autonomous Driving of FDCT, Macau SAR. Dr. Xu's main research interests lie in parallel and distributed computing and cloud computing, in particular, with an emphasis on resource management for system's performance, reliability, availability, power efficiency, and security, and in big data and data-driven intelligence applications in smart city and self-driving vehicles. He has published more than 200 papers in journals and conferences. He serves on a number of journal editorial boards, including the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, and China Science Information Sciences. He is a fellow of the IEEE.



Dejing Dou received the bachelor's degree from Tsinghua University, China, in 1996 and the Ph.D. degree from Yale University in 2004. He is the Head of Big Data Laboratory, Baidu Research, Beijing, China. He is currently on the sabbatical leave from his Professor position in the Computer and Information Science Department at the University of Oregon, Eugene, OR, USA, where he also leads the Advanced Integration and Mining (AIM) Lab. He also serves as the Director of the NSF IUCRC Center for Big Learning (CBL).

His research areas include artificial intelligence, data mining, data integration, information extraction, and health informatics. Dejing Dou has published more than 100 research papers, some of which appear in prestigious conferences and journals like AAAI, IJCAI, KDD, ICDM, ACL, EMNLP, CIKM, ISWC, JIIS and JoDS. His DEXA'15 paper received the best paper award. His KDD'07 paper was nominated for the best research paper award. He is on the Editorial Boards of Journal on Data Semantics, Journal of Intelligent Information Systems, and PLOS ONE. He has been serving as program committee members for various international conferences and as program co-chairs for four of them. Dejing Dou has received over \$5 million PI research grants from the NSF and the NIH. He was promoted to Full Professor in 2016.