CASCADED CONTEXT DEPENDENCY: AN EXTREMELY LIGHTWEIGHT MODULE FOR DEEP CONVOLUTIONAL NEURAL NETWORKS

Xu Ma, Zhinan Qiao*, Jingda Guo, Sihai Tang, Qi Chen, Qing Yang, ✉ Song Fu*

Department of Computer Science and Engineering
University of North Texas, Denton, Texas, USA

ABSTRACT

In this paper, we present a cascaded context dependency module, which is a highly lightweight module that can improve the performance of deep convolutional neural networks for various visual tasks. Inspired by the feature pyramid work in object detection and the context dependency work in image recognition, we consider to cascade the contexts of multi-scale feature maps to aggregate the locality and globality in a local region. We further extract the dependency between original input and cascaded contexts for feature recalibration. Without employing learnable layers, our method introduces almost no additional parameters and computations. Furthermore, Our module can be seamlessly plugged into many existing CNN architectures to improve the performance. Experiments on ImageNet and MS COCO benchmarks indicate that our method can achieve results on par with or better than related work. Qualitatively, we achieve an absolute 1.42% (77.3137% *vs.* 75.8974%) top-1 classification accuracy improvement based on ResNet50 on ImageNet 2012 validation set with negligible computational overhead. Besides, our method yields significant gains on the MS COCO benchmark for the object detection task. All codes and models are made publicly available [1].

***Index Terms*—** CNN, lightweight, context dependency, multi-scale

1. INTRODUCTION

Capturing the properties of locality and globality concurrently is a central importance in deep neural networks [1, 2]. In computer vision, CNNs achieve this by stacking a series of convolutional layers sequentially, enlarging the receptive field from small to large [3].

While convolutional layers operate the features in a local receptive field, they consider no communication with other features that are not in the range of receptive field. To meet the requirement of globality, a CNN architecture has to progressively and recursively propagate features in a large or even global range using multiple convolutional layers. That is, the shallow layers consider the features more locally, whereas the deep layers infer feature more globally [4]. Unambiguously, this is a departure from the primary conception that locality and globality should be captured concurrently. Furthermore, the final extracted global concepts may bias toward the most informative local features, and differ from the essential global concepts.

In this paper, we propose cascaded context dependency module, a simple yet effective module that captures the feature dependency on both local region and global range. Motivated by the feature dependency work in [1, 5], we calculate the dependencies between the original features and extracted context. An apparent distinction is that our extracted context is obtained from different scales of the original feature maps using pooling operations. By doing so, we naturally integrate the contexts from different scales together. We consider three different scales of contexts: fine, coarse, and global, which are easily achieved by adaptively average pooling operations. The three contexts are then merged to achieved our cascaded contexts, aggregating locality and globality naturally.

A second property of our method is the transformation operation on the feature dependency. Rather than leveraging learnable layers to transform the dependencies, we follow the work in normalization layers [6, 7, 8] that utilize the normalization statistics, but operated in groups. This approach in our module exhibits considerably simpler designs and fewer computations than [1, 5] while efficiently process the obtained dependency. Concurrent with our work, some efforts [9, 10, 11] also considers normalization statistics for different visual tasks. However, they are different from us essentially.

We evaluate our module on ImageNet 2012 [12] image classification and MS COCO [13] object detection tasks, compared with a range of related work. Consistently, our method yields a systematic improvement with negligible computational overhead. Integrating our cascaded context dependency module into base CNN architectures, we surpass the original counterpart by a large margin and achieve a performance that on par with or even better than related

* Equal contribution
✉ Corresponding author
[1] We submit all the codes, pre-trained models, and training log files to https://github.com/13952522076/ParameterFree.

insertable modules. Our proposed module is designed as an external module independent of the original networks, making it easy to be integrated with various existing CNN architectures. Experimental results have demonstrated that our module can improve the performances of various CNN architectures, and for different vision tasks.

2. RELATED WORK

Context Dependency. Features not only respond in a local region, but may also own a long-range feature dependency [1, 14]. In pursuit of long-range feature dependency, Non-Local Network [1] introduces a non-local block, which is a general non-local operation calculating the dependency for each pair of features. However, the pairwise computation is heavy. Recently, [5] simplified the non-local operation by exploiting the dependency between original feature maps and the global context. We also notice that some attention modules [15, 16] consider the context dependency inherently.

Multi-scale aggregation. Multi-scale aggregation holds prevalence for a long time. A most distinct example in CNNs is the spatial pyramid pooling [17]. To eliminate the limitation of fixed input size of earlier CNNs, [17] proposed a spatial pyramid pooling strategy that is adaptively pooling the feature maps to three scales and feed to the finial fully-connected layers. The simple but effective strategy dramatically improves the CNN capability for image classification and object detection tasks. Similar to [17], Feature Pyramid Network [18], Cascaded R-CNN [19] and Multi-scale Network [20] also consider the use of multi-scale intermediate feature maps for low-level/high-level feature semantics fusion and feature map resolution compensation. In this paper, we follow previous efforts and leverage the multi-scale aggregation for our local and global context fusion.

Normalization. Normalization layers [6, 8, 7] have been an integral part of recent deep neural networks. Inspired by Batch Normalization [6], a series of normalization algorithms have been proposed. To overcome the problem of small batch size in Batch Normalization, Layer Normalization [7] normalizes all channels for each sample and avoids the dependency on a batch of samples. Group Normalization [21] divides channels into several groups and normalizes each group individually. Inspired by these work, we also employ normalization statistics in our module.

3. METHOD

We present the Cascaded Context Dependency Module (CCD module in short), which is a plug-in module for improving the presentation ability of various CNN architectures.

Suppose the input feature map of a CCD module is $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$, where C denotes the channel number and $H \times W$ means the spatial resolution. We divide the input $\mathbf{x}$ to g groups along channel dimension, that is $\mathbf{x} = [x_1, x_2, \ldots, x_g]$, and perform the cascaded context dependency for each group independently and parallelly. For each group $x_i \in \mathbb{R}^{\frac{C}{g}\times H\times W}$, we extract the cascaded context $c_i \in \mathbb{R}^{\frac{C}{g}\times H\times W}$ using three adaptive average pooling operations. Next we calculate the dependency between x_i and c_i and normalize it to facilitate training. The resulting output will be resized to a range of $(0, 1)$ to re-calibrate the original x_i. Generally, the process of our CCD module can be summarized as:

$$\mathbf{y} = cat\left(\forall_{i\in g} sig\left(\mathcal{N}\left(f((x_i, c_i))\right) \otimes x_i\right), \tag{1}$$

where "cat" indicates the concatenation operations; "sig" denotes sigmoid activation function; "$\mathcal{N}$" means normalization; $f(\cdot)$ is the dependency function and " $\otimes$ " denotes element-wise multiplication. In the following subsections, we will present each component in detail.

3.1. Cascaded Context

We conduct our cascaded context c_i using three scales of the original x_i. To achieve this, we adopt the average pooling operation to obtain three scales of resolution: 4×4, 2×2, and 1×1. Notice that the 1×1 scale is the result of global average pooling, making the resulting output contains a global context of x_i. Likewise, the other two contain the local context, fine or coarse, of different scales of patches. All three outputs are enlarged using the nearest interpolation to match the size of $H \times W$. We next aggregate the three outputs by summation to get the cascaded context $c_i \in \mathbb{R}^{\frac{C}{g}\times H\times W}$. The resulting cascaded context c_i is divided by 3 to match the magnitude of the original feature map. Remarkably, we eschew the use of learnable layers for aggregation due to the reason of efficiency and computational complexity. We emphasize that while learnable parameters will certainly achieve better cascaded context, but the computational burdens increase accordingly. The ablation study on different scale strategies is presented in Section 4.2.

3.2. Feature Dependency

Following the recent work [1, 25] on feature dependency, we formulate the pairwise dependency function as:

$$f\left(x_i^p, c_i^p\right) = {x_i^p}^{\mathsf{T}} c_i^p, \tag{2}$$

where p is the geometric position and $x_i^p, c_i^p \in \mathbb{R}^{\frac{C}{g}}$ is the corresponding features. Some other dependency functions can also be exploited, like cosine similarity function:

$$f\left(x_i^p, c_i^p\right) = \frac{{x_i^p}^{\mathsf{T}} c_i^p}{max\left(\left\|x_i^p\right\|_2 * \left\|c_i^p\right\|_2, \varepsilon\right)}, \tag{3}$$

where ε is a small value (we set ε to 10^{-8}) to avoid division by zero. However, the exploitation of pairwise dependency function is not the primary concern in our paper. We employ Equation 2 as our dependency function by default.

Model	top-1 acc.	top-5 acc.	FLOPs (G)	Parameters (M)
ResNet50 [22]	75.8974	92.7224	4.122	25.557
SE-ResNet50 [15]	77.2877$_{(\uparrow 1.39)}$	**93.6478**	4.130	28.088
GE-ResNet50 [23]	76.2357$_{(\uparrow 0.34)}$	92.9847	**4.127**	**25.557**
CBAM-ResNet50 [24]	77.2840$_{(\uparrow 1.39)}$	93.6005	4.139	28.090
SK-ResNet50 [16]	**77.3657**$_{(\uparrow 1.47)}$	93.5256	4.187	26.154
GC-ResNet50 [5]	74.8966$_{(\downarrow 1.00)}$	92.2812	4.130	28.105
CCD-ResNet50 (ours)	**77.3137**$_{(\uparrow 1.42)}$	**93.6489**	**4.122**	**25.560**

Table 1: Comparison results of single-crop classification accuracy (%) and complexity on the ImageNet validation set. The best two performances are marked in **bold**.

3.3. Normalization

Suppose the resulting dependency between x_i and c_i in group i is $\mathbf{D} \in \mathbb{R}^{H\times W}$, we compute the mean μ and standard deviation σ by

$$\mu = \frac{1}{H\times W}\sum_{i=1}^{H\times W}\mathbf{D}_i, \sigma = \left(\frac{1}{H\times W}\sum_{i=1}^{H\times W}(\mathbf{D}_i-\mu)^2\right)^{\frac{1}{2}}. \tag{4}$$

We normalize the dependency $\mathbf{D}$ over the spatial dimension, and apply an affine function to avoid distribution shifting. The normalized dependency can be formulated as :

$$\mathbf{D} = \lambda\frac{\mathbf{D}-\mu}{\sigma+\epsilon}+\beta, \tag{5}$$

where ϵ is a small constant (10^{-5}) for numerical stability, λ and β are a pair of learnable parameters.

3.4. Efficiency

In pursuit of efficiency, our cascaded context dependency module mainly relies on non-learnable operations. During the whole process, we only introduce g pairs of λ and β. In all our experiments, we set the group number g as 64, unless otherwise noted. The elegant design makes our module extremely lightweight. For example, when applying our module to the commonly used ResNet50 architecture, we only introduce 2k (that is $16\times 64\times 2$) parameters. Clearly, it is negligible compared with the original parameter number (25.56M) in ResNet50.

4. EXPERIMENTS

Besides efficiency, our proposed module is also powerful. In this section, we systematically evaluate CCD module on ImageNet 2012 [12] and MS COCO [13] benchmarks for image recognition and object detection tasks respectively. Extensive experimental results reveal the effectiveness of our proposed method.

4.1. Image Classification on ImageNet

We first compare our method with several other related modules on the ImageNet dataset. ImageNet contains 1.28M training images, and 50K validation images belong to 1000 predefined categories. We train all models follow standard practice in [22, 26]. All training images are randomly cropped to 224×224 and horizontally flipped at a possibility of 50%. We train all models using synchronous SGD with momentum 0.9 and weight decay 1e-4. The learning rate is initiated as 0.1 and decreased by a factor of 10 every 30 epochs. We train from scratch for 100 epochs on a server with 8 Tesla V100 GPUs, and each GPU has 32 images in a mini-batch (256 in total). All experiments are implemented using PyTorch framework [27]. For comparison, we compare our method with several SOTAs, including SENet [15], SKNet [16], GENet [23], CBAM [24], etc. For fairness, we integrate all these modules to ResNet50 and report the top-1 and top-5 classification accuracy in Table 1.

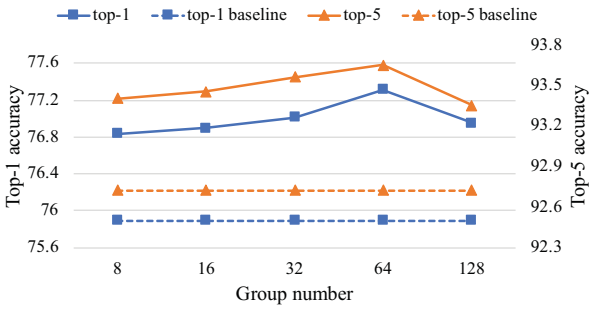

Fig. 1: The influence of group number on CCD module. Best viewed in color.

Table 1 shows that integrating our proposed module with the vanilla ResNet50 consistently improves the performance by **1.42%** top-1 classification accuracy, whereas almost no additional computations are introduced. When compared with other state-of-the-art modules, like SE, CBAM, SK, *etc.*, we achieve a result than on par with or even better the all these modules but introduces much fewer FLOPs and parameters. This phenomenon indicates that our proposed module is powerful, demonstrating the efficacy of the cascaded context dependency that generates richer global and local representational dependencies between features.

4.2. Ablation Study

To study the intrinsic properties of our method, we conduct a series of ablation studies to disentangle the influence of each detail design on our cascaded context dependency module.

Group number *g*. We evaluate the group number g in the range of $[8, 16, 32, 64, 128]$. Figure 1 depicts the classification curve under different number of groups in our module. Intuitively, with the

Strategy	top-1	top-5
baseline	75.8974	92.7224
$[1, 4, 8]$	77.0840	93.5897
$[1, 2, 4]^*$	76.9371	93.4195
$[1, 2, 4]$	77.3137	93.6489

(a) The performance of different implementations of cascaded context.

function	top-1	top-5
baseline	75.8974	92.7224
Eq. [2]	77.3137	93.6489
Eq. [3]	77.1959	93.4563

(b) The performance of different implementations of cascaded context.

sigmoid		✓	✓	✓
dependency			✓	✓
cascaded				✓
normalize				✓
top-1	75.90	75.91	76.79	77.31
top-5	92.72	92.80	93.35	93.65

(c) The ablation study of each components.

Table 2: Ablation studies on ImageNet validation set. We report the top-1 and top-5 accuracy (%) based on ResNet50. "*" indicates that we remove the 4×4 in conv4_x and further remove the 2×2 in the conv5_x in ResNet50.

Detector	Backbone	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GMac	Parameters(M)
RetinaNet	ResNet50 [22]	36.2	55.9	38.5	19.4	39.8	48.3	239.32	37.74
RetinaNet	SE-ResNet50 [15]	37.4	57.8	39.8	20.6	40.8	50.3	239.43	40.25
RetinaNet	CCD-ResNet50 (ours)	**37.8**	**58.5**	**40.1**	**21.6**	**41.5**	**50.9**	239.32	37.74
Cascade R-CNN	ResNet50 [22]	40.6	58.9	44.2	22.4	43.7	54.7	234.71	69.17
Cascade R-CNN	GC-ResNet50 [5]	41.1	59.7	44.6	23.6	44.1	54.3	234.82	71.69
Cascade R-CNN	CCD-ResNet50 (ours)	**42.5**	**61.1**	**46.4**	**24.7**	**45.9**	**56.5**	234.71	69.17

Table 3: Object detection performance (%) on the MS-COCO validation dataset. The best ones are highlighted in "**bold**".

increase of group number g, the top-1 accuracy also increases and appears to saturate at the g value of 64. As the value of g continues to increase, the performance gain decreases. Similar phenomenon can also be observed in top-5 accuracy. Empirically, we set g to 64 in our module.

Cascaded context. The central philosophy of our module can be distilled into how to capture the locality and globality in a feature map simultaneously. Driven by the success of using multi-scale feature maps, we exhibit a considerably simple design that cascade different scales of feature maps to get the cascaded context. Here, we evaluate the influence of different scale strategies in our module and report the results in Table 2a. Notably, the cascaded context dependency module can always improve the performance of base CNN architecture by a large margin, regardless of the specific strategy of the cascaded context. Besides, we also notice that keeping the three scales in all stages can achieve the best performance, indicating that even the deep layers require the collaboration of globality and locality.

Dependency function. Table 2b compares two dependency functions in our CCD module. The improvements of these two are similar, just a tiny fluctuation of less than 0.12% (77.3137% *vs.* 77.1959%). This observation indicates that our module is not sensitive to the selection of feature dependency functions, suggesting that the performance gain comes from the design rather than a certain dependency instance.

Components ablation. We also present a detailed ablation study on each component in Table 2c. In line with our expectations, all the ablative variants of CCD module are instrumental, improving the performance of the original ResNet50 by a large margin. Combining all these, we arrive at the best performance 77.31%.

4.3. Object Detection on MS COCO

We further evaluate our method for object detection on the MS COCO benchmark to showcase the compatibility of our method on other visual tasks. We train all models on the 80k training set and evaluate on the 40k validation set. All models are trained for 24 epochs with pre-trained backbones directly taken from Table 1. Following the common practice [28], we report the mean Average-Precision (mAP) for the detection performance. We employ RetinaNet [28] and Cascade R-CNN [19] as detectors with a ResNet-50 (and corresponding variants) backbone in the detection experiments.

Without increasing the complexity, our CCD module improves RetinaNet by 1.6% mAP and Cascade R-CNN by 1.9% mAP. Likewise, our module consistently performs better than others (SE and GC) by a large margin, with even fewer parameters and FLOPs. In particular, our method outperforms SE module by 2.2% mAP (19.4% *vs.* 21.6%) on small objects. The promising results on object detection task show that our CCD module is conducive to improve the performance of various visual tasks significantly.

5. CONCLUSION

In this paper, we present a new method, Cascaded Context Dependency module (CCD in short), which captures the locality and globality in the feature maps concurrently and efficiently. For this purpose, we exploit the feature dependencies between original features and cascaded context. Our proposed method is gracefully designed and computationally lightweight. We experimentally show the significant improvements of our method for image recognition and object detection tasks. Moreover, CCD is an insertable module that that be easily integrated with various CNN architectures.

6. ACKNOWLEDGEMENT

This research was supported in part by NSF grants (CNS-1852134 and ECCS-2010332) and a grant from Fujitsu Laboratories of America, Inc.. We also thank Google for providing access to the Google Cloud Platform.

7. REFERENCES

[1] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le, "Attention augmented convolutional networks," *arXiv preprint arXiv:1904.09925*, 2019.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[4] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang, "FishNet: A versatile backbone for image, region, and pixel level prediction," in *NIPS*, 2018, pp. 754–764.

[5] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu, "GCNet: Non-local networks meet Squeeze-Excitation networks and beyond," *arXiv preprint arXiv:1904.11492*, 2019.

[6] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[9] Xiang Li, Xiaolin Hu, and Jian Yang, "Spatial group-wise enhance: Improving semantic feature learning in convolutional networks," *arXiv preprintarXiv:1905.09646*, 2019.

[10] Ruan Dongsheng, Wen Jun, and Zheng Nenggan, "Linear context transform block," *arXiv preprint arXiv:1909.03834*, 2019.

[11] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam, "Srm: A style-based recalibration module for convolutional neural networks," in *ICCV*, 2019, pp. 1854–1862.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[15] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-Excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[16] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang, "Selective kernel networks," in *CVPR*, 2019, pp. 510–519.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *TPAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[19] Zhaowei Cai and Nuno Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018, pp. 6154–6162.

[20] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*, 2016, pp. 354–370.

[21] Yuxin Wu and Kaiming He, "Group normalization," in *ECCV*, 2018, pp. 3–19.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[23] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi, "Gather-Excite: Exploiting feature context in convolutional neural networks," in *NIPS*, 2018, pp. 9401–9411.

[24] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "CBAM: Convolutional block attention module," in *ECCV*, 2018, pp. 3–19.

[25] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin, "Local relation networks for image recognition," *arXiv preprint arXiv:1904.11491*, 2019.

[26] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 1492–1500.

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.