GDC block: gradient-guided direction-aware convolution block for image classification

Jinye Peng1, Maomei Liu1*, Lei Tang2, Sheng Zhong1 and Hangzai Luo1

1School of Information and Technology, Northwest University, Shaanxi, 710127, China
2Xi'an Microelectron Technology Institute, Shaanxi, 710127, China
*Corresponding author’s e-mail: maomeiliu@stumail.nwu.edu.cn

Abstract. Convolutional Neural Network (CNN) has achieved great success in visual applications. In the field of image classification, researchers usually customize CNN models to meet the needs of different real-world applications. It consumes a lot of human labor and computing resources but only achieves slight performance improvement. Besides, some recent works try to integrate prior knowledge into classic CNN models to improve their accuracy, but they are usually only effective for special applications. In this paper, we propose a gradient-guided direction-aware convolution (GDC) block. It can be used to replace low-level convolutions of existing CNN without changing the off-the-shelf architecture. The gradient priors provide object shapes that CNN's low-level convolution requires. And the direction-aware mechanism expands the receptive field size. This scheme is a trade-off between model size and model accuracy. Experimental results show it can moderately reduce the size of any CNN models while enhancing their performance.

1. Introduction
CNN has achieved great success in visual applications. Especially in recent years, the performance of image classification has been significantly improved, and some classic models have become the mainstream architecture (e.g., VGG [1] and ResNet [2]). In order to deploy CNN in real-world applications, researchers generally design new architectures to meet different requirements. This consumes a lot of computing resources and human labor, on the contrary, only provides a little effect on performance improvement. Besides, these well-designed models generally improve accuracy through more trainable parameters and complex connections. However, due to the limited computing resources of terminal devices, CNN must provide high accuracy within a certain computing budget. Complex models are often difficult to deploy. Therefore, without the need to customize the new architecture at a high price, we consider that it makes sense to improve the efficiency of existing classic CNN models.

2. Efficient models
Recently, many efficient models have been proposed. SqueezeNet [3] reduces parameters and computation significantly. MobileNet [4] builds lightweight deep neural networks based on a streamlined architecture that uses depthwise separable convolution. They focus on modifying the model structure to reduce the model parameters, thus improve model efficiency. However, this usually comes at the cost of reduced accuracy. Alternatively, some methods work on training small networks. Distillation [5] uses a larger network to teach a smaller network. Neural Architecture Search (NAS) [6]
searches model structure automatically. However, these methods require a large number of computing resources to obtain a small model (generally, they consume thousands of GPU hours), which is often unacceptable in many application scenarios.

Alternatively, some researches focus on how to effectively integrate priors into CNN to improve its performance. A hybrid model that couples discrete wavelet transforms (WT) and artificial neural networks (ANN) is proposed for forecasting water temperature in [7]. Besides, a graph wavelet neural network (GWNN) is presented in [8], which leverages graph wavelet transform to address the shortcomings of previous spectral graph CNN methods that depend on graph Fourier transform. However, due to the lack of interpretability of CNN, these models are usually only effective for specific applications.

3. GDC block

In this paper, we propose the GDC block to improve the model efficiency without changing off-the-shelf architectures. Our GDC block can be easily used to replace the low-level convolution of existing networks. By effectively improving the low-level feature extraction efficiency, it can moderately reduce the size of any CNN models, while enhancing their performance. In the following introduction, we present the GDC block and describe its role in existing CNN models. The off-the-shelf network that employs the GDC block is called GDCNet. For convenience, we take the top left corner of the image as the origin and establish a rectangular coordinate system with the horizontal direction as the X-axis and the vertical direction as the Y-axis.

3.1. GDC block structure

Figure 1 gives the detail of the GDC block. In our GDC block, an original convolution is split into two branches. One branch keeps the original filters to extract basic features while the other branch explores object shapes with a direction-aware mechanism. It utilizes gradient priors to explore object shapes, and with the consists of X-axis and Y-axis one-dimensional filters, it can further achieve direction-aware convolution. Specifically, we first calculate the gradient $xG$ and $yG$ of the input feature, then we use X-axis filters to aware horizontal direction information on $G_x$, and adopt Y-axis filters to aware vertical direction information on $G_y$. Taking $3\times3$ convolutions as an example, the GDC block contains three parallel layers with kernel sizes of $3\times3$, $1\times3$, and $3\times1$. The $3\times3$ output and the summarized-output of $1\times3$ and $3\times1$ are concatenated as the output of the GDC block. Without changing a given architecture, we simply replace the low-level convolutional layer with a GDC block to construct GDCNet.

Figure 1. Giving an example: here we have three convolution layers with a filter size of $3\times3$, $1\times3$, and $3\times1$, respectively. They take different input. $3\times3$ convolution takes original input while $1\times3$ takes $G_x$ and $3\times1$ takes $G_y$ as input. Finally, we concatenate $3\times3$ convolution output and the sum of $1\times3$ and $3\times1$ convolution outputs as the block's output. Especially, each branch has half the number of filters as the original.
3.2. Why GDC block works

The effectiveness of our GDC block can be summarized as the following three aspects.

First of all, gradient is one of the most important low-level image features that can be used to improve classification accuracy. This information can effectively detect edges and further present object shapes. Generally, we use the following formula to calculate the X-axis and Y-axis gradients:

\[
G_x = \begin{bmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1
\end{bmatrix} * I
\]

\[
G_y = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
+1 & +2 & +1
\end{bmatrix} * I
\]

Where \( I \) denotes input features. The above convolution formula can be realized by simple addition and subtraction operations, which does not occupy network parameters and only consumes very little computing resources. Besides, through visualization of well-trained CNN models, researchers find that CNN's low-level feature maps show promising evidence of extracting low-level image features. Hence, without changing the off-the-shelf architecture, we present the GDC block to replace its low-level convolution.

Second, we find that \( G_x \) represents the pixel change along the X-axis, and the X-axis convolution on \( G_x \) may expand the receptive field size along X-axis. And the Y-axis convolution is the same. It can be simply expressed as:

\[
f(G_x) = f(f_x(F),W)
\]

\[
f(G_y) = f(f_y(F),W)
\]

Where \( f \) presents network filters with trainable parameters \( W \), \( f_x \) and \( f_y \) indicates gradient filters. Generally, enlarging the receptive field size can be considered as a strategy to improve CNN performance. Hence, compared with the original convolution, this direction-aware mechanism can improve the classification accuracy of existing architectures.

Third, thanks to the two branch structure, the GDC block not only keeps extracting original basic features but also reduces the number of parameters.

4. Experimental analysis

We experiment GDC block with several representative benchmark models including Cifar-quick [9], VGG, ResNet, and DenseNet [10] on CIFAR-10, CIFAR-100 [11], and ImageNet [12]. Cifar-quick only has 3 convolution layers that can be seen as a representative of existing small models. VGG and ResNet are the most widely used classical CNN models in all vision tasks. As a result, it is more convincing to use them as benchmark models. We only replace the low-level convolution of the above networks with the GDC block. Since Cifar-quick only has three convolution layers, we consider they are all low-level convolution. And we take the convolution layers before the first downsample operation in VGG, ResNet, and DenseNet as low-level convolution because they can be regarded as extracting low-level features and their feature maps have the same resolution as the input image.

| Model      | Baseline Top-1 (%) | GDCNet Top-1 (%) | Top-1 ↑ (%) |
|------------|--------------------|------------------|-------------|
| Cifar-quick| 86.24              | 87.36            | 1.12        |
| VGG        | 94.12              | 94.50            | 0.38        |
| ResNet-56  | 94.31              | 95.07            | 0.76        |
| DenseNet-40| 94.29              | 94.92            | 0.63        |
As can be observed from Table 1 and Table 2, the performance of all models is consistently lifted by a clear margin, suggesting that the benefits of the GDC block can be combined with various architectures. Obviously, with the GDC block, the efficiency of the low-level features improves. Besides, the results clearly show Cifar-quick gets a significant accuracy improvement. It confirms that with the GDC block, the existing small models can perform more efficiently on end devices.

Table 2. Top-1 accuracy of GDCNet and the normally trained baselines on CIFAR-100.

| Model     | Baseline Top-1 (%) | GDCNet Top-1 (%) | Top-1 ↑ (%) |
|-----------|--------------------|------------------|-------------|
| Cifar-quick | 53.22              | 58.64            | 5.42        |
| VGG       | 74.56              | 76.20            | 1.64        |
| ResNet-56 | 73.58              | 73.76            | 0.18        |
| DenseNet-40 | 73.14             | 73.52            | 0.38        |

To further verify the efficiency of our GDC block, we experiment GDCNet and the normally trained deep models on ImageNet. As shown in Table 3, GDCNet still achieves better performance. This stresses the GDC block can improve the performance of existing CNN models.

Table 3. Top-1 accuracy of GDCNet and the normally trained baselines on ImageNet.

| Model     | Baseline Top-1 (%) | GDCNet Top-1 (%) | Top-1 ↑ (%) |
|-----------|--------------------|------------------|-------------|
| AlexNet   | 55.92              | 57.68            | 1.76        |
| ResNet-18 | 70.36              | 71.54            | 1.18        |
| DenseNet-121 | 75.15          | 76.06            | 0.91        |

Table 4 presents the model parameters of GDCNet and baselines. It can be clearly seen that GDCNet consumes fewer model parameters than its baseline architecture. For a small model, all its convolutions are replaced by GDC block, thus the parameters reduce a lot. For deep models, only some low-level convolutions are replaced by GDC block, thus the parameters reduce slightly. However, while maintaining model architecture and improving performance, it can also reduce some parameters, we consider that the GDC block has made significant progress.

Table 4. Model paramenters of GDCNet and the baselines.

| Model     | Baseline params (M) | GDCNet params (M) | Params ↓ (%) |
|-----------|--------------------|------------------|-------------|
| Cifar-quick | 0.47               | 0.35             | 25.53       |
| VGG       | 59.1               | 57.2             | 3.21        |
| ResNet-56 | 3.41               | 3.20             | 6.16        |

5. Conclusion
We propose a novel GDC block to explore the effectiveness of integrating gradient priors into CNN models. Through the direction-aware mechanism, it can improve the performance of several benchmark models. Given an off-the-shelf network, the GDC block is a trade-off between model size and model accuracy. Since there is no need to redesign the model architecture, it is meaningful for real-world applications.

Acknowledgments
This work was supported by “135 HGJ” 2018ZX01012101-007 Radiation-immune IC microsystem.

References
[1] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[2] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778.
[3] Iandola, F.N., Han, S., Moskewicz, M.W., et al. (2016). SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. arXiv preprint arXiv:160207360.
[4] Howard, A.G., Zhu, M., Chen, B., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861.
[5] Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531.
[6] Liu, H., Simonyan, K., Yang, Y. (2018). Darts: Differentiable architecture search. arXiv preprint arXiv:180609055.
[7] Graf, R., Zhu, S., Sivakumar, B. (2019). Forecasting river water temperature time series using a wavelet-neural network hybrid modelling approach. Journal of Hydrology, 578: 124115.
[8] Xu, B., Shen, H., Cao, Q., et al. (2019). Graph wavelet neural network. arXiv preprint arXiv:190407785.
[9] Snoek, J., Larochelle, H., Adams, R.P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In: Advances in neural information processing systems. pp. 2951-2959.
[10] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700-4708.
[11] Krizhevsky, A., Hinton, G. (2009). Learning Multiple Layers of Features from Tiny Images. Handbook of Systemic Autoimmune Diseases, 1(4).
[12] Deng, J., Dong, W., Socher, R., et al. (2009) Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp: 248-255.