Paper Note

"GhostNet: More Features from Cheap Operations"

Paper Link: CVPR 2020

Lisen Dai

Nov 8, 2020

1 Background

1.1 Problem

Traditional CNNs usually need a large number of parameters and floating point operations (FLOPs) to achieve a satisfactory accuracy.

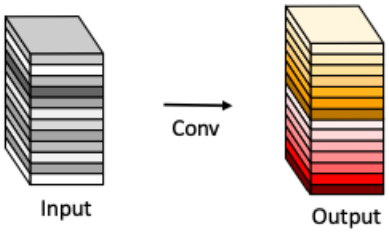
1.2 Previous Work

1. Network pruning.
2. Low-bit quantization.
3. Knowledge distillation.
4. Efficient structure design.

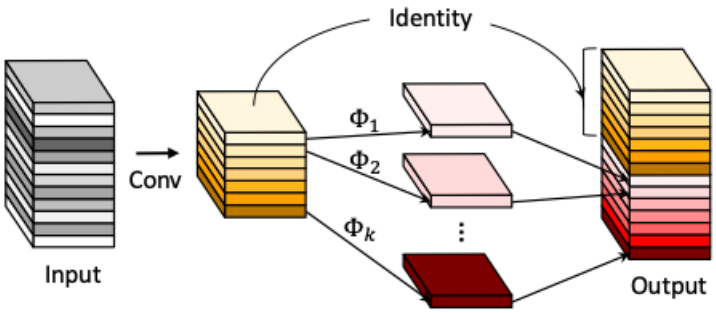
- MobileNet utilized the depthwise and pointwise convolutions to construct a unit for approximating the original convolutional layer with larger filters and achieved comparable performance.
- ShuffleNet further explore a channel shuffle operation to enhence the performance of lightweight models.

2 Contributions

2.1 Ghost Module



(a) The convolutional layer.



(b) The Ghost module.

Figure 2. An illustration of the convolutional layer and the proposed Ghost module for outputting the same number of feature maps. Φ represents the cheap operation.

It is about to change the following equation for CNN layers:

$$Y = X * f + b$$

to,

$$Y' = X * f'$$

where $f' \in \mathcal{R}^{c \times k \times k \times m}$ is the utilized filters, and for Y' here goes,

$$y_{ij} = \Phi_{i,j}(y'_i), \forall i = 1, ..., m, j = 1, ..., s,$$

where y'_i is the i-th intrinsic feature map in Y', $\Phi_{i,j}$ in the above function is the j-th (except the last one) linear operation for generating the j-th ghost feature map y_{ij}.

2.2 Applications

2.2.1 Ghost Bottlenecks

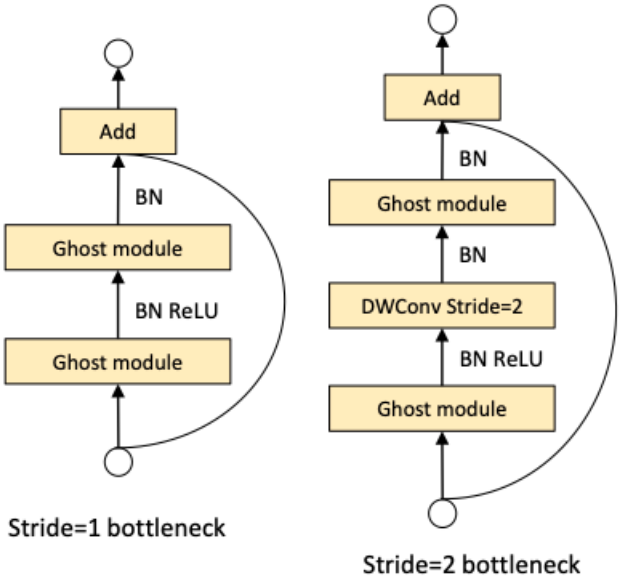


Figure 3. Ghost bottleneck. Left: Ghost bottleneck with stride=1; right: Ghost bottleneck with stride=2.

In practice, the primary convolution in Ghost module here is pointwise convolution for its efficiency.

2.2.2 GhostNet

Use the structure of MobileNetV3 but re- place the bottleneck block in MobileNetV3 with our Ghost bottleneck.
Also, there is a kind of GhostNet with width multiplier α called GhostNet-$\alpha\times$. α here is a factor on the number of channels uniformly at each layer, To customize the network for smaller and faster models or higher accuracy on specific tasks.
Width multiplier can control the model size and the computational cost quadratically by roughly α^2. Usually smaller α leads to lower latency and lower performance, and vice versa.

3 Experiments

3.1 setup

dataset: CIFAR-10, ImageNet ILSVRC 2012 dataset, MS COCO object detection benchmark.

3.2 Visualization of Feature Maps

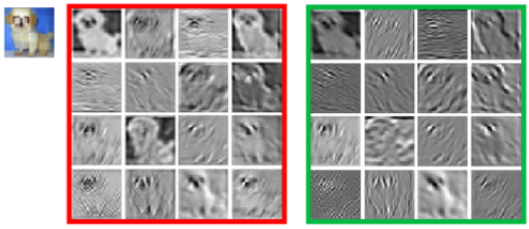


Figure 4. The feature maps in the 2nd layer of Ghost-VGG-16. The left-top image is the input, the feature maps in the left red box are from the primary convolution, and the feature maps in the right green box are after the depthwise transformation.

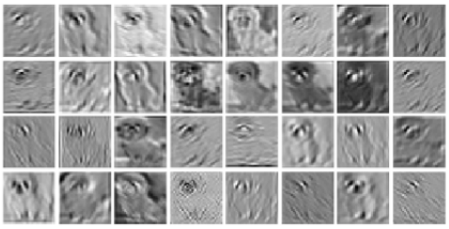


Figure 5. The feature maps in the 2nd layer of vanilla VGG-16.

3.3 Comparison

3.3.1 CIFAR-10

Table 5. Comparison of state-of-the-art methods for compressing VGG-16 and ResNet-56 on CIFAR-10. - represents no reported results available.

Model	Weights	FLOPs	Acc. (%)
VGG-16	15M	313M	93.6
ℓ_1-VGG-16 [31, 37]	5.4M	206M	93.4
SBP-VGG-16 [18]	-	136M	92.5
Ghost-VGG-16 (s=2)	7.7M	158M	**93.7**
ResNet-56	0.85M	125M	93.0
CP-ResNet-56 [18]	-	63M	92.0
ℓ_1-ResNet-56 [31, 37]	0.73M	91M	92.5
AMC-ResNet-56 [17]	-	63M	91.9
Ghost-ResNet-56 (s=2)	0.43M	63M	**92.7**

3.3.2 ImageNet

Table 6. Comparison of state-of-the-art methods for compressing ResNet-50 on ImageNet dataset.

Model	Weights (M)	FLOPs (B)	Top-1 Acc. (%)	Top-5 Acc. (%)
ResNet-50 [16]	25.6	4.1	75.3	92.2
Thinet-ResNet-50 [39]	16.9	2.6	72.1	90.3
NISP-ResNet-50-B [59]	14.4	2.3	-	90.8
Versatile-ResNet-50 [49]	11.0	3.0	74.5	91.8
SSS-ResNet-50 [23]	-	2.8	74.2	91.9
Ghost-ResNet-50 (s=2)	13.0	2.2	**75.0**	**92.3**
Shift-ResNet-50 [53]	6.0	-	70.6	90.1
Taylor-FO-BN-ResNet-50 [41]	7.9	1.3	71.7	-
Slimmable-ResNet-50 0.5× [58]	6.9	1.1	72.1	-
MetaPruning-ResNet-50 [36]	-	1.0	73.4	-
Ghost-ResNet-50 (s=4)	6.5	1.2	**74.1**	**91.9**

3.3.3 Visual Benchmarks

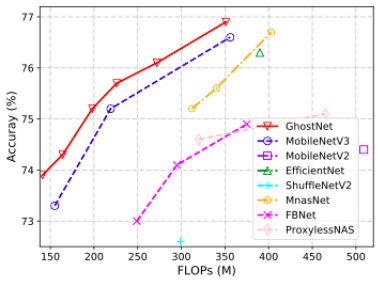


Figure 6. Top-1 accuracy *v.s.* FLOPs on ImageNet dataset.

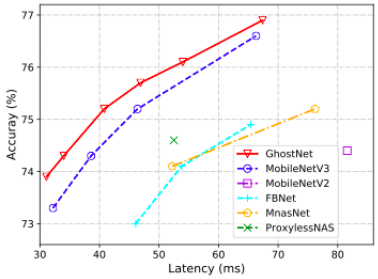


Figure 7. Top-1 accuracy *v.s.* latency on ImageNet dataset.

Table 6. Comparison of state-of-the-art methods for compressing ResNet-50 on ImageNet dataset.

Model	Weights (M)	FLOPs (B)	Top-1 Acc. (%)	Top-5 Acc. (%)
ResNet-50 [16]	25.6	4.1	75.3	92.2
Thinet-ResNet-50 [39]	16.9	2.6	72.1	90.3
NISP-ResNet-50-B [59]	14.4	2.3	-	90.8
Versatile-ResNet-50 [49]	11.0	3.0	74.5	91.8
SSS-ResNet-50 [23]	-	2.8	74.2	91.9
Ghost-ResNet-50 (s=2)	13.0	2.2	**75.0**	**92.3**
Shift-ResNet-50 [53]	6.0	-	70.6	90.1
Taylor-FO-BN-ResNet-50 [41]	7.9	1.3	71.7	-
Slimmable-ResNet-50 0.5× [58]	6.9	1.1	72.1	-
MetaPruning-ResNet-50 [36]	-	1.0	73.4	-
Ghost-ResNet-50 (s=4)	6.5	1.2	**74.1**	**91.9**

3.3.4 MS COCO

Table 8. Results on MS COCO dataset.

Backbone	Detection Framework	Backbone FLOPs	mAP
MobileNetV2 1.0× [44]	RetinaNet	300M	26.7%
MobileNetV3 1.0× [20]		219M	26.4%
GhostNet 1.1×		164M	26.6%
MobileNetV2 1.0× [44]	Faster R-CNN	300M	27.5%
MobileNetV3 1.0× [20]		219M	26.9%
GhostNet 1.1×		164M	26.9%