Grouping Bilinear Pooling

Rui Zeng
zengrui@mail.ustc.edu.cn
Jingsong He
hjss@ustc.edu.cn

University of Science and Technology of China
Hefei, China

Abstract

Fusion of the extracted high-order features by capturing complex correlation between features to obtain a better representation performs well in visual tasks. As a simple and effective high-order feature interaction representation, the bilinear representation has achieved remarkable results in many visual tasks: fine-grained image classification, semantic segmentation and so on. However, bilinear pooling has not been widely used due to the bilinear representation up to hundreds of thousands or even millions of dimensions. In this paper, we propose grouping bilinear pooling (GBP) that the representation captured by GBP can achieve the same effect with less than 0.4% parameters compare with full bilinear representation. This extreme compact representation largely overcomes the high redundancy of the full bilinear representation, the computational cost and storage consumption. It can be used as a plug-and-play module with convenient operation. Comparing with other state-of-the-art approaches, it achieves competitive performance. The effectiveness of the proposed GBP is proved by experiments on the widely used fine-grained recognition datasets.

1 Introduction

Convolutional neural network (CNN) has been widely used in various computer vision tasks such as image classification [18, 31], object detection [24, 29] and semantic segmentation [2, 30]. The key is that CNN can extract rich semantic features through the stacked convolutional layers and the elaborate design of structure. To make better use of the extracted semantic features, lots of meaningful and enlightening works are proposed to get better feature representation.

Most studies obtain the feature representation of input image by pooling high-order features [37, 38], making model to pay attention on valuable information [12, 32] or aggregating the features of different levels [21, 30], then apply it to subsequent tasks. Besides, some other studies use the high-order statistical information of features to obtain better feature representation, such as VLAD [11], Fisher Vector [8, 27], spatial pyramids [19] and bilinear pooling [23, 36]. Among them, the full bilinear pooling (FBP) [23] captures the complex association between paired features and uses the bilinear representation for classification which makes remarkable achievements in fine-grained image classification. However, the bilinear representation is as high as hundreds of thousands or even millions of dimensions. It is far higher than the aggregating representation, resulting in huge computational load and memory consumption, and limiting the expansion of model structure.

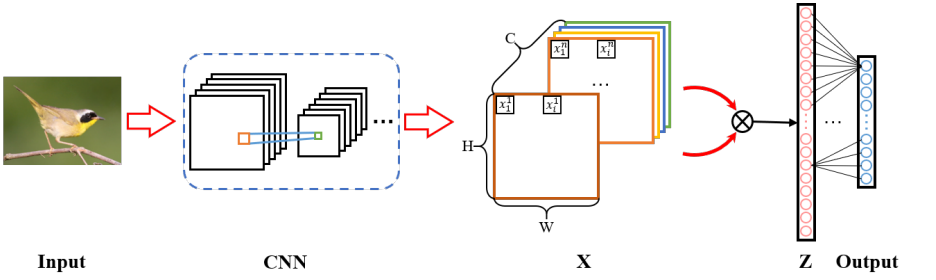

Figure 1: The architecture of Bilinear CNN

In order to reduce the dimensions of bilinear representation, [5, 14, 16, 35, 39] simplify the bilinear operations in different ways. [5] regards the image classification based on bilinear pooling as a linear kernel machine and proves that bilinear pooling enabled the linear classifier to have the discriminating ability of a second-order kernel machine. Then, Random Maclaurin (RM) [13] and Tenor Sketch (TS) [28] are used for low-dimensional approximation, and a compact bilinear pooling (CBP) is established. [14] uses Hadamard product to perform low-dimensional approximation bilinear pooling. [16, 35] also reduces the dimensions by carrying out low-dimensional feature mapping to get a low-rank representation. Learnable grouping module is introduced for semantic grouping in [39]. It also reduces the dimensions of bilinear representation to a certain extent.

Inspired by various compression methods, we note that further compression is possible for bilinear representation. According to the analysis of bilinear pooling in Section 2, it is shown intuitively that the bilinear representation is actually a low-rank self-correlation and cross-correlation representation. Then, the reason for high redundancy of bilinear representation is analyzed. To compress the bilinear representation to the extreme, we propose grouping bilinear pooling (GBP) to minimize the dimensions of bilinear representation with only 0.4% parameters of full bilinear representation. With the fewest parameters, GBP can reach the best accuracy among compact methods. It also achieves competitive performance comparing with other state-of-the-art approaches. Experiments on the widely used datasets CUB-200-2011 [34] and Stanford Cars [17] show the effectiveness of GBP.

2 Analysis of Bilinear Pooling

In FBP [23], in order to get the bilinear representation Z of input image, the image will pass through the convolutional neural network first to obtain the high-order feature representation $X, X \in \mathbb{R}^{C\times H\times W}$, where C is the number of feature layers, the height and the width of feature layers are H and W. For each feature layer, there are $H \times W$ different locations. The network architecture is illustrated in Figure 1.

Here, we define local descriptor $x_i^T = \left[x_i^1, x_i^2, \ldots, x_i^C\right] \in \mathbb{R}^C (i \in [1, \ldots, HW])$, x_i^n represents the value of ith position on the nth feature layer. Besides, for convenience, considering x_i as a vector which dimension is C. The bilinear representation Z is as follow:

$$Z = \frac{1}{HW}\sum_i^{HW} x_i x_i^T = \frac{1}{HW}\begin{bmatrix} \sum_i^{HW} x_i^1 x_i^1 & \ldots & \sum_i^{HW} x_i^1 x_i^C \\ & \ldots & \\ \sum_i^{HW} x_i^1 x_i^C & \ldots & \sum_i^{HW} x_i^C x_i^C \end{bmatrix} \tag{1}$$

Vectorizing Z, $vector\ (Z) \in \mathbb{R}^{C^2}$. Assuming C is 512, $vector\ (Z)$ will be up to $250K$ dimensions. The high dimensions of bilinear representation result in high computation and storage costs.

The representation Z is used for classification after passing through the full-connection layer,

$$Out\,put = ZW_C + b = \frac{1}{HW}\left(\sum_i^{HW} x_i x_i^T\right) W_C + b_C \tag{2}$$

Where $W_C \in \mathbb{R}^{C^2 \times N}$ is the weight matrix of full-connection layer, $b_C \in \mathbb{R}^k$, $Out\,put \in \mathbb{R}^N$, N is the number of categories. In general, $C^2 \gg N$, the rank of W_C is as follow:

$$rank\,(W_C) \leq \min\left(C^2, N\right) = N \tag{3}$$

Vectorizing the ith feature layer: $f_i^T = \left[x_1^i, x_2^i, \ldots, x_{HW}^i\right] \in \mathbb{R}^{HW}$, $F^T = [f_1, f_2, \ldots, f_C] \in \mathbb{R}^C$,

$$Z = \frac{1}{HW}\begin{bmatrix} f_1^T f_1 & \cdots & f_1^T f_c \\ & \cdots & \\ f_1^T f_c & \cdots & f_c^T f_c \end{bmatrix} = \frac{1}{HW} F F^T \tag{4}$$

$$Out\,put = ZW_C + b = \frac{1}{HW}\left(FF^T\right) W_C + b_C \tag{5}$$

In [5], RM [13] is used to sample the feature layers, then bilinear pooling is carried out. In fact, the representation obtained by [5] is the recombination of part of the element in Z. The representation obtained by [14] using the low-dimensional approximation of Hadamard product is the elements on the diagonal of Z. This low-rank approximation actually abandons the vast majority of information of Z. Losing of information is inevitable, although the dimensions of representation is reduced.

X contains C different feature layers f, the bilinear representation Z in Equation 1 is a symmetric matrix. The elements on the diagonal of Z are the dot product sum of the corresponding positions of feature layer itself. The scalar obtained by point-wise product can be regarded as the pixel-level self-correlation of the feature layer to some extent. Similarly, the elements on the non-diagonal of Z are the dot product sum of the corresponding positions of different feature layers, which can be regarded as the cross-correlation between feature layers.

Since the bilinear representation obtained by bilinear pooling is a correlation representation with extremely high dimensions between high-order feature layers (increasing with the square of the number of feature layers), it will greatly increase the parameters to be learned by the full-connection layers even there is only a single-layer full-connection layer. Furthermore, there is nearly half of the calculations in symmetric matrices Z are repeated, obviously, it greatly reduces computational efficiency and results in redundancy of the model.

3 Grouping Bilinear Pooling

According to Equation 3, hoping full-connection layer to establish high-efficient contact between Z and $Out\,put$ is impractical. Thus, minimizing the huge gap of dimensions between Z and $Out\,put$ is a highly cost-effective way.

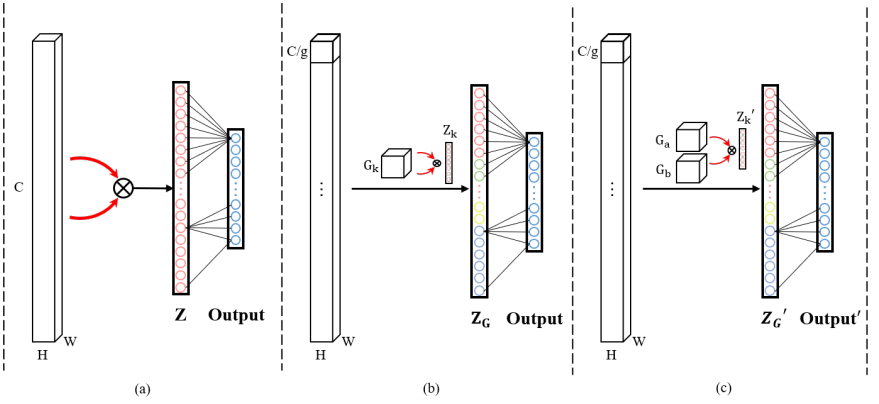

Figure 2: (a) is FBP, bilinear pooling is performed in pairs in all feature layers. (b) is the intra-group bilinear pooling, bilinear pooling performs in each grouped feature layer group. (c) is the inter-group bilinear pooling, bilinear pooling performs between two different grouped feature layer groups.

We propose grouping bilinear pooling (GBP) that by grouping the feature layers X and performing intra-group bilinear pooling (Intra-GBP) or strongly constrained inter-group bilinear pooling (Inter-GBP), the information of the original bilinear representation can be greatly preserved and the model can be extremely compressed. Figure 2 shows the difference between FBP [23] and GBP.

3.1 Intra-group Bilinear Pooling

Dividing F into g groups. Noting that the maximum of g is C while carrying out Intra-GBP. For the kth group $G_k^T = \left[f_1, f_2, \ldots, f_{C/g}\right], G_k \in \mathbb{R}^{\frac{C}{g}}$,

$$Z_k = \frac{1}{HW} G_k G_k^T \tag{6}$$

$$Z_G = Concat\left(Z_1, Z_2, \ldots, Z_k\right) \tag{7}$$

$$Output = Z_G W_G + b_G \tag{8}$$

Where $vector\left(Z_k\right) \in \mathbb{R}^{\left(\frac{C}{g}\right)^2}, k \in [1,2,\ldots,g], Z_G \in \mathbb{R}^{\frac{C^2}{g}}, W_G \in \mathbb{R}^{\frac{C^2}{g} \times N}, b_G \in \mathbb{R}^N$.

For FBP, when the full bilinear representation Z is used for classification, the parameters that the full-connection layer needs to learn are as high as C^2N. While for the grouping bilinear operation proposed by us, the parameters that the full-connection layer needs to learn are C^2N/g.

$$rank\left(W_G\right) \leq \min\left(\frac{C^2}{g}, N\right) \tag{9}$$

When g is small, this bilinear operation after grouping still requires large computational resources (Intra-GBP at $g = 1$ is equivalent to FBP), and the dimension of Intra-GBP is still

too high. As g gets bigger and bigger, it will bring huge benefits. The influence of changing g will be explained in the experiment section.

The representation Z_k in Equation 6 has similar properties to the representation Z obtained by FBP, that is, Z_k is also a symmetric matrix and there is still computational redundancy. Thus, we propose inter-group bilinear pooling for further improvement.

3.2 Inter-group bilinear pooling

Same as Intra-GBP, dividing F into g groups. The difference is that bilinear pooling is carried out between two different grouped feature layers groups G_a and G_b ($G_a, G_b \in \mathbb{R}^{\frac{C}{g}}, G_a \neq G_b$) in Inter-GBP. G_a and G_b are from grouped g groups, and each group is selected only once.

$$Z_k{}' = \frac{1}{HW} G_a G_b^T \tag{10}$$

$$Z_G{}' = Concat\left(Z_1{}', Z_2{}', \ldots, Z_k{}'\right) \tag{11}$$

$$Output' = Z_G{}' W_G{}' + b_G{}' \tag{12}$$

Where $vector\left(Z_k{}'\right) \in \mathbb{R}^{\left(\frac{C}{g}\right)^2}, k \in [1, 2, \ldots, g/2], Z_G{}' \in \mathbb{R}^{\frac{C^2}{2g}}, W_G{}' \in \mathbb{R}^{\frac{C^2}{2g} \times N}, b_G{}' \in \mathbb{R}^N$.

Similar to Intra-GBP, Inter-GBP will yield huge benefits when g is large enough. Besides, g should be a multiple of 2 due to the specific group selection method of Inter-GBP, the maximum and minimum of g are $C/2$ and 2. The full connection layer needs to learn the parameters: $C^2N/2g$.

$$rank\left(W_G{}'\right) \leq \min\left(\frac{C^2}{2g}, N\right) \tag{13}$$

Furthermore, in order to simplify the operation of Inter-GBP, feature layers are grouped in sequence, and the groups for bilinear pooling are selected in order.

4 Experiment

4.1 Datasets, Backbone and Experiment Configurations

Datasets

We conduct experiments on two widely used fine-grained image classification datasets: CUB [34] and Stanford Cars [17]. In all experiments, we only used the category labels of images. The details of datasets are shown in Table 1.

Dataset	Training	Testing	Category
CUB [34]	5994	5794	200
Stanford Cars [17]	8144	8041	196

Table 1: Datasets Details

Backbones

In order to compare with compact bilinear pooling methods and the states-of-the-art approaches using different methods, we use VGG-16 [31], ResNet-50 [9], ResNet-101 [9] and

ResNet-152 [9] pretrained on the ImageNet [4] image classification dataset as our backbone networks respectively (removing full-connection layers, using the GBP pooling layer and the new full-connection layer instead).

Experimental Configurations

The experiment was carried out on the server of Ubuntu system, using PyTorch [26] framework and 4 NVIDIA GTX 1080*Ti* GPUs for distributed model training. The size of input image is 448×448 and our data augmentation follows the commonly used methods. During the training, the pre-training weight of the model on ImageNet [4] was first loaded and frozen, and the parameters of the full-connection layer between GBP representation and outputs were fine-tuned. During the fine-tuning, the initial learning rate was 0.0003, and the Adam optimizer [15] was adopted with factor=0.2, patience=3, cooldown=3. After fine-tuning the epoch to 45 epochs, the model was unfreezing, then the learning rate was adjusted to 0.0001. In all experiments, the same experimental configurations were followed.

4.2 Evaluation

First, we used VGG-16 [31] as the backbone network to perform Intra-GBP and Inter-GBP on CUB [34] dataset and Stanford Cars [17] dataset respectively. Then, after grouping bilinear pooling, the obtained representation was used to classification directly by full-connection layer. The high-order feature layer X extracted from VGG-16 has 512 feature channels, which was divided into g groups, $\mathrm{g} \in [1,2,4,8,16,32,64,128,256]$. Noting that the Intra-GBP degenerates to FBP [23] at $g = 1$, and the minimum of g is 2 in Inter-GBP.

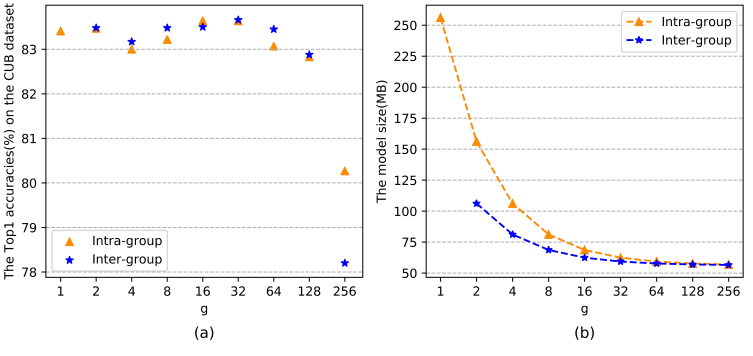

Figure 3: (a) is the Top1 accuracy of Intra-GBP and Inter-GBP based on VGG-16 on CUB dataset; (b) is the corresponding model size (including backbone CNN).

The original FBP achieved an accuracy of 84.01% on CUB dataset, and our reimplementation achieved an accuracy of 83.43%. The experiment results of Intra-GBP and Inter-GBP on the CUB dataset are shown in Figure 3 and Figure 4. Intra-GBP(Inter-GBP) achieved the best accuracy 83.64%(83.66%) at $g = 32$ on CUB, 91.42%(92.49%) at $g = 64$ on Stanford Cars. In the same grouping case, Inter-GBP tends to mean higher performance and fewer parameters.

As discussed, Intra-GBP still has inherent redundancy, and Inter-GBP can reduce the redundancy. Comparing with the Intra-GBP, the Inter-GBP is always more compressive and performs better with same backbone. This characteristic does not vary with backbone

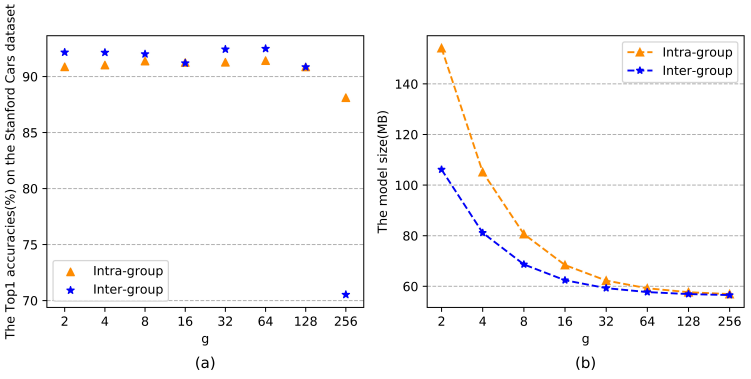

Figure 4: (a) is the Top1 accuracy of Intra-GBP and Inter-GBP based on VGG-16 on Stanford Cars dataset; (b) is the corresponding model size (including backbone CNN).

and dataset. Therefore, we only show Inter-GBP with better performance in subsequent experiments.

In order to further verify the effectiveness of GBP with different backbones, we also used ResNet-50, ResNet-101 and ResNet-152 as the backbone to perform GBP respectively. With a more powerful backbone, GBP performs better. Table 2 shows the performances of Inter-GBP based on ResNet-50 on CUB and Stanford Cars.

The groups	2	4	8	16	32	64	128	256	512	1024
CUB(%)	83.79	84.19	85.13	85.28	85.23	85.11	85.07	85.21	85.32	**85.54**
Model size(*MB*)	987.81	497.81	297.81	197.81	147.81	122.81	110.31	104.06	100.94	99.37
Stanford Cars(%)	92.11	92.34	92.35	92.29	92.33	92.49	92.61	92.75	92.74	**92.86**
Model size(*MB*)	971.81	489.81	293.81	195.81	146.81	122.31	110.06	103.94	100.87	99.34

Table 2: The Inter-GBP experiments on CUB dataset and Stanford Cars dataset (ResNet-50)

In the case of g=1024, the best accuracy of Inter-GBP on CUB dataset(Stanford Cars dataset) reaches 85.54%(92.86%), which is 2%(0.4%) higher than Inter-GBP based on VGG-16. When g goes from 2 to 1024, the model size goes from 987.81*MB*(971.81*MB*) to 99.37*MB*(99.34*MB*) and the accuracy goes from 83.76%(92.11%) to 85.54%(92.86%).

Noting that the accuracies reach the highest at g=1024 (maximum number of grouping). It is because the Inter-GBP representation is the most compact when maximum number of grouping is used. In all experiments with ResNet as the backbone, more groupings often represent better performance.

4.3 Comparison with Compact Bilinear Pooling Methods

It is similar to the comparison method in [16], we compare our GBP with some classical compact bilinear pooling methods [5, 16, 22, 23] in details.

Assuming that the categories to be classified is N, and bilinear pooling is carried out on the feature layers with the size of $c \times h \times w$, where c is the feature channels, the height and width of feature layers are h and w (VGG-16: h=w=28, c=512; ResNet-50: h=w=14, c=2048). For more intuitive comparison, taking the experiment on CUB as an example. The

input size of the images is 448×448. The configurations of the comparative experiment are as follows: g=128(VGG-16), g=1024(ResNet-50), m=100, r=8, d=8192, N=200.

Backbone	Method	Dimension	Computing		Parameters		
			Pooling	Classifying	Projection	Classifier	Total
VGG-16	FBP [23]	$c^2[262K]$	$O(hwc^2)$	$O(Nc^2)$	0	Nc^2	$200MB$
	iFBP [23]	$c^2[262K]$	$O(hwc^2)$	$O(Nc^2)$	0	Nc^2	$200MB$
	CBP-TS [5]	$d[10K]$	$O(hw(c+dlogd))$	$O(Nd)$	$2c$	Nd	$8MB$
	CBP-RM [5]	$d[10K]$	$O(hwcd)$	$O(Nd)$	$2cd$	Nd	$48MB$
	LRBP-I[16]	$mhw[78K]$	$O(hwcm)$	$O(Nrmhw)$	cm	Nrm	$0.8MB$
	LRBP-II[16]	$m^2[10K]$	$O(hw(cm+m^2))$	$O(Nrm^2)$	cm	Nrm	$0.8MB$
	Intra-GBP(*ours*)	$c^2/g[2K]$	$O(hwc^2/g^2)$	$O(Nc^2/g)$	0	Nc^2/g	$1.6MB$
	Inter-GBP(*ours*)	$c^2/2g[1K]$	$O(hwc^2/2g^2)$	$O(Nc^2/2g)$	0	$Nc^2/2g$	$0.8MB$
ResNet-50	FBP [23]	$c^2[4194K]$	$O(hwc^2)$	$O(Nc^2)$	0	Nc^2	$3200MB$
	Intra-GBP(*ours*)	$c^2/g[4K]$	$O(hwc^2/g^2)$	$O(Nc^2/g)$	0	Nc^2/g	$3.2MB$
	Inter-GBP(*ours*)	$c^2/2g[2K]$	$O(hwc^2/2g^2)$	$O(Nc^2/2g)$	0	$Nc^2/2g$	$1.6MB$

Table 3: Comparison of different compact bilinear pooling methods. We used VGG-16 and ResNet-50 as the backbone respectively to compare the computational complexity, representation dimensions and the parameters need to be learned (excluding the backbone network).

The detailed comparisons of different compact bilinear pooling methods are shown in Table 3. The comparison contents include feature dimension, computational complexity and the number of parameters. Under the same configurations, the performances of compact bilinear pooling methods based on VGG-16 are shown in Table 4.

Dataset	FBP [23]	iFBP [23]	CBP-TS [5]	CBP-RM [5]	LRBP[16]	Intra-GBP(ours)	Inter-GBP(ours)
CUB(%)	84.01	**85.80**	84.00	83.86	84.21	83.64	83.66
Stanford Cars(%)	91.18	92.10	90.19	89.54	90.92	91.42	**92.49**

Table 4: The performances of different compact bilinear pooling methods on CUB dataset and Stanford Cars dataset. (Based on VGG-16)

According to the comparisons, GBP is more competitive than other compact methods in almost all the aspects. When using VGG-16 as backbone, the representation dimensions of Inter-GBP($g = 128$) are only 0.4% of FBP, and the calculation amount of pooling is reduced by 4 orders of magnitude. Besides, the parameters need to learn (excluding the backbone network) are reduced by 99.6%. Comparing with [5, 16], the grouping operation in GBP is performed without learning additional parameters. Assuming ResNet-50 is used as backbone, the bilinear feature representations based on FBP [23](GBP) will reach $4194K$ ($2K$), and the parameters of model will reach $3200MB$ ($1.6MB$).

From the performances, GBP greatly reduces the number of model parameters and achieves or closes to the best accuracies of other compact methods. On Stanford Cars dataset, the Inter-GBP based on VGG-16 reaches the best accuracy of 92.49%. When using a more powerful backbone, GBP will achieve all-round transcendence.

4.4 Comparison with the state-of-the-art

Generally, it is hard to balance accuracy and the complexity of model when bilinear pooling is applied. With the extreme compression, GBP is able to using more powerful backbones to improve performance. We embedded GBP in different backbones and compared it with other methods. The performances of baselines, full bilinear pooling based methods, com-

pact bilinear pooling based methods, GBP(ours) methods and other state-of-the-art methods relating to channels are shown in Table 5.

Method		Backbone	Dimension	Parameters	CUB(%)	Stanford Cars(%)
Baselines	VGG-16 [30]	-	25K	20MB	74.59	85.05
	ResNet-50 [9]	-	2K	1.6MB	82.15	92.19
	ResNet-101 [9]	-	2K	1.6MB	82.58	92.56
	ResNet-152 [9]	-	2K	1.6MB	82.74	92.64
Full Bilinear Pooling	FBP [23]	VGG-16	260K	200MB	84.01	91.18
	iFBP [22]				85.80	**92.10**
	MoNet-FBP [8]				**86.40**	91.80
Compact Bilinear Pooling	CBP [5]	VGG-16	10K	8MB	84.00	90.19
	LRBP [16]		10K	8MB	84.21	**90.90**
	MoNet-TS [8]		10K	8MB	**85.70**	90.80
	FBC [1]		8K	6.4MB	84.30	-
	SBP-EN [20]		10K	8MB	84.50	**90.90**
State-of-the-art	SWP [10]	VGG-16	-	-	-	90.70
		ResNet-50	-	-	-	92.30
		ResNet-101	-	-	-	93.10
	HBPASM [33]	ResNet-34	-	-	86.80	92.80
	HBP [35]	VGG-16	24K	19MB	87.01	93.70
	SEF [25]	VGG-16	-	-	81.10	88.30
		ResNet-50	-	-	87.30	94.00
	MC-loss [1]	ResNet-50	-	-	87.30	93.70
	CIN [6]	ResNet-50	-	-	87.50	94.10
		ResNet-101	-	-	**88.10**	**94.50**
GBP(ours)	**Inter-GBP**	VGG-16	1K	0.8MB	83.66	92.49
		ResNet-50	2K	1.6MB	85.54	92.86
		ResNet-101	2K	1.6MB	86.10	93.76
		ResNet-152	2K	1.6MB	**86.31**	**94.22**

Table 5: The performances of different methods on CUB dataset and Stanford Cars dataset. From top to bottom, the five blocks respectively list baselines, full bilinear pooling based methods, compact bilinear pooling based methods, other state-of-the-art methods relating to channels and our method.

Comparing with the methods based on bilinear pooling and compact bilinear pooling, GBP is the most compact method and achieves the best performance. Especially, Inter-GBP improves the best performance of compact bilinear pooling from 91.80% to 94.22% on Stanford Cars dataset.

In addition, due to lack of simplicity and convenience, few papers apply compact bilinear pooling approaches to high-performing backbones. And we also tried to perform other compact BP on better backbones, but the result is not good enough. GBP compresses the bilinear representation to the extreme so that it can be used with more powerful backbones to achieve competitive performance.

The grouping operation of GBP is performed at the channel level. Comparing with other state-of-the-art methods relating to channels [1, 6, 10, 25, 33, 35], GBP shows performance as good as or even better than these methods.

Spatially weighted pooling (SWP) [10] strategy was proposed to improve the robustness and effectiveness of the feature representation. [33] devised a novel model Hierarchical Bilinear Pooling with Aggregated Slack Mask (HBPASM) to generate a RoI-aware image feature representation for better performance. [35] first proposed to obtain a better feature representation by adjusting channel dimensions and performing Hadamard product between different hierarchical feature layers. Mutual-channel loss [1] achieved the state-of-the-art performance when implemented on top of common base networks. Channel permutation

and weighted combination regularization in [25] also shown its effectiveness. And channel interaction network [6] allowed the model to learn the complementary features from the correlated channels, yielding stronger fine-grained representation. These methods achieved the state-of-the-art in different ways.

Compared with SWP [10], which also uses VGG-16, ResNet-50 and ResNet-101 as backbones, GBP has better performance with different backbone. GBP outperformed most other state-of-the-art methods with the accuracy of 94.22% on the Stanford Cars dataset, only 0.3% lower than the best method [6].

5 Conclusions

We propose GBP in this paper, then Intra-GBP and Inter-GBP is introduced. By bilinear pooling the grouped high-order feature layers in different way, GBP greatly reduces the dimensions of the bilinear representation, the storage consumption and calculations. Comparing with other compact bilinear methods, GBP achieves the-state-of-the-art. Meanwhile, the experiments show that GBP has also achieved competitive performance comparing with other state-of-the-art approaches.

With the reduction of bilinear representation dimensions brought by GBP, bilinear pooling can be applied in other visual tasks efficiently. And GBP can be embedded into different models as a plug-and-play module with convenient operation. We believe GBP has much more potential. In the future work, we plan to further explore GBP by combining it with other methods.

References

[1] Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Transactions on Image Processing*, 2020.

[2] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[3] K. Daniilidis, P. Maragos, and N. Paragios. Improving the fisher kernel for large-scale image classification. *Eccv*, 2010.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Yu Gao, Xintong Han, Xun Wang, Weilin Huang, and Matthew Scott. Channel interaction networks for fine-grained image categorization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[7] Zhi Gao, Yuwei Wu, Xiaoxun Zhang, Jindou Dai, Yunde Jia, and Mehrtash Harandi. Revisiting bilinear pooling: A coding perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 04 2020.

[8] Mengran Gou, Fei Xiong, Octavia Camps, and Mario Sznaier. Monet: Moments embedding network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] Qichang Hu, Huibing Wang, Teng Li, and Chunhua Shen. Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[11] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.

[12] H. Jie, S. Li, S. Gang, and S. Albanie. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2017.

[13] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. *Journal of Machine Learning Research*, 22:583 – 591, 2012.

[14] Jin-Hwa Kim, Kyoung On, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 10 2016.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[16] S. Kong and C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *IEEE Computer Society*, pages 7025–7034, 2017.

[17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25(2), 2012.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[20] Q. Liao, D. Wang, H. Holewa, and M. Xu. Squeezed bilinear pooling for fine-grained visual categorization. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2020.

[21] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] Tsung-Yu Lin and Subhransu Maji. Improved bilinear pooling with cnns. 07 2017.

[23] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457, 2015.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *European Conference on Computer Vision*, 2016.

[25] Wei Luo, Hengmin Zhang, Jun Li, and Xiu-Shen Wei. Learning semantically enhanced feature for fine-grained image classification. *IEEE Signal Processing Letters*, 2020.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[27] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.

[28] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Computer Vision & Pattern Recognition*, 2016.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Springer, Cham*, 2015.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[32] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In *Computer Vision – ECCV 2018*, 2018.

[33] Min Tan, Guijun Wang, Jian Zhou, Zhiyou Peng, and Meilian Zheng. Fine-grained classification via hierarchical bilinear pooling with aggregated slack mask. *IEEE Access*, 2019.

[34] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds200-2011 dataset. *Advances in Water Resources - ADV WATER RESOUR*, 07 2011.

[35] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You. Hierarchical bilinear pooling for fine-grained visual recognition. 2018.

[36] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. In *Computer Vision – ECCV 2018*, 2018.

[37] D. Yu, H. Wang, P. Chen, and Z. Wei. Mixed pooling for convolutional neural networks. In *International Conference on Rough Sets & Knowledge Technology*, 2014.

[38] Matthew Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[39] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Learning deep bilinear transformation for fine-grained image representation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.