

双线性汇合(bilinear pooling)在细粒度图像分析及 其他领域的进展综述

原创： 张皓

*SIGAI*推荐

[SIGAI 资源大汇总](#)

作者简介：

张皓

南京大学计算机系机器学习与数据挖掘所（LAMDA）

研究方向：计算机视觉和机器学习（视觉识别和深度学习）

个人主页：goo.gl/N715YT

细粒度图像分类旨在同一大类图像的确切子类。由于不同子类之间的视觉差异很小，而且容易受姿势、视角、图像中目标位置等影响，这是一个很有挑战性的任务。因此，类间差异通常比类内差异更小。双线性汇合（bilinear pooling）计算不同空间位置的外积，并对不同空间位置计算平均汇合以得到双线性特征。外积捕获了特征通道之间成对的相关关系，并且这是平移不变的。双线性汇合提供了比线性模型更强的特征表示，并可以端到端地进行优化，取得了和使用部位（parts）信息相当或甚至更高的性能。

在本文，我们将对使用双线性汇合进行细粒度分类的方法发展历程予以回顾。研究方向大致分为两类：设计更好的双线性汇合过程，以及精简双线性汇合。其中，对双线性汇合过程的设计主要包括对汇合结果规范化过程的选择及其高效实现，以及融合一阶和二阶信息。精简双线性汇合设计大致有三种思路：利用PCA降维、近似核计算、以及低秩双线性分类器。此外，双线性汇合的思想也被用于其他计算机视觉领域，例如风格迁移、视觉问答、动作识别等。我们也将介绍双线性汇合在这些领域的应用。

1. 数学准备

在本节，我们介绍本文用要用到的符号和在推导时需要的数学性质。

深度描述向量(deep descriptor) $x_i \in \mathbb{R}^D$ 。其中 $1 \leq i \leq HW$ 。例如对VGG-16网络，我们通常使用relu5-3层的特征提取图像的深度描述向量，那么 $H=W=14, D=512$ 。

描述矩阵 (descriptor matrix) $X \in \mathbb{R}^{D \times HW}$ 。定义为

$$\mathbf{X} := [\mathbf{x}_1 \ \mathbf{x}_1 \ \cdots \ \mathbf{x}_{HW}]$$

均值向量 (mean vector) $\mu \in \mathbb{R}^D$ 。定义为

$$\mu := \frac{1}{HW} \sum_{i=1}^{HW} \mathbf{x}_i = \frac{1}{HW} \mathbf{X} \mathbf{1}$$

格拉姆矩阵 (Gram matrix) $G \in \mathbb{R}^{D \times D}$ 。定义为

$$\mathbf{G} := \frac{1}{HW} \sum_{i=1}^{HW} \mathbf{x}_i \mathbf{x}_i^\top = \frac{1}{HW} \mathbf{X} \mathbf{X}^\top$$

协方差矩阵 (covariance matrix) $\Sigma \in \mathbb{R}^{D \times D}$ 。定义为

$$\Sigma := \frac{1}{HW} \sum_{i=1}^{HW} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top = \frac{1}{HW} \mathbf{X} \tilde{\mathbf{I}} \mathbf{X}^\top$$

其中

$$\tilde{\mathbf{I}} := \mathbf{I} - \frac{1}{HW} \mathbf{1} \mathbf{1}^\top \in \mathbb{R}^{HW \times HW}$$

是中心化矩阵 (centering matrix)。

分数向量 (score vector) $\mathbf{y} \in \mathbb{R}^k$, softmax层的输入, k 是分类任务的类别数。

由于双线性汇合相关的论文涉及许多矩阵运算, 尤其是迹运算。如下性质在推导时将有帮助。这些性质在使用时即可以从左向右使用, 也可以从右向左使用。

向量化操作的迹等价

$$(\text{vec } \mathbf{A})^\top (\text{vec } \mathbf{B}) = \text{tr } \mathbf{A}^\top \mathbf{B} = \text{tr } \mathbf{A} \mathbf{B}^\top$$

弗罗贝尼乌斯范数 (Frobenius norm) 的迹等价

$$\|\mathbf{A}\|_F^2 = \text{tr } \mathbf{A}^\top \mathbf{A}$$

矩阵迹满足交换率和结合率

$$\text{tr } \mathbf{A} \mathbf{B} = \text{tr } \mathbf{B} \mathbf{A}, \text{tr } (\mathbf{A} + \mathbf{B}) = \text{tr } \mathbf{A} + \text{tr } \mathbf{B}$$

矩阵转置不改变迹

$$\text{tr } \mathbf{A}^\top = \text{tr } \mathbf{A}$$

矩阵幂等价于特征值分别取幂

$$\mathbf{A}^c = \mathbf{U} \Sigma^c \mathbf{V}^\top$$

对函数分 $f(x,y)$ ，双线性 (bilinear) 是指当固定其中一个参数 (例如 x) 时, $f(x,y)$ 对另一个参数 (例如 y) 是线性的。在这里，研究的双线性函数是形如 $f(x,y)=x^\top A y$ 这样的形式。本文关注的双线性汇合叫双线性这个名字是受历史的影响，在早期两个分支是不同的，现在主流做法是两个分支使用相同的输入，整个操作将变为非线性而不是双线性，但这个名称沿用至今。

2. 双线性汇合

双线性汇合在深度学习复兴前就已经被提出，随后，在细粒度图像分类领域得到了广泛使用。本节将介绍双线性汇合及其变体。

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. ICCV 2015: 1449-1457.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear convolutional neural networks for fine-grained visual recognition. TPAMI 2018, 40(6): 1309-1322.

双线性汇合操作通过计算深度描述向量的格拉姆矩阵 G 捕获特征通道之间成对的相关关系。随后，将格拉姆矩阵展成向量

$$\mathbf{s} := \text{vec } \mathbf{G} \in \mathbb{R}^{D^2}$$

并进行规范化 (normalization)

$$s_k \leftarrow \text{sign } s_k \sqrt{|s_k|}, \quad \mathbf{s} \leftarrow \frac{\mathbf{s}}{\|\mathbf{s}\|}$$

得到最终的双线性特征。

在原文中，Lin等人使用了两个不同的网络得到双线性汇合的不同分支，动机是希望一个分支学到位置 (where) 信息，而另一个分支学到外观 (what) 信息。事实上，这是不可能的。

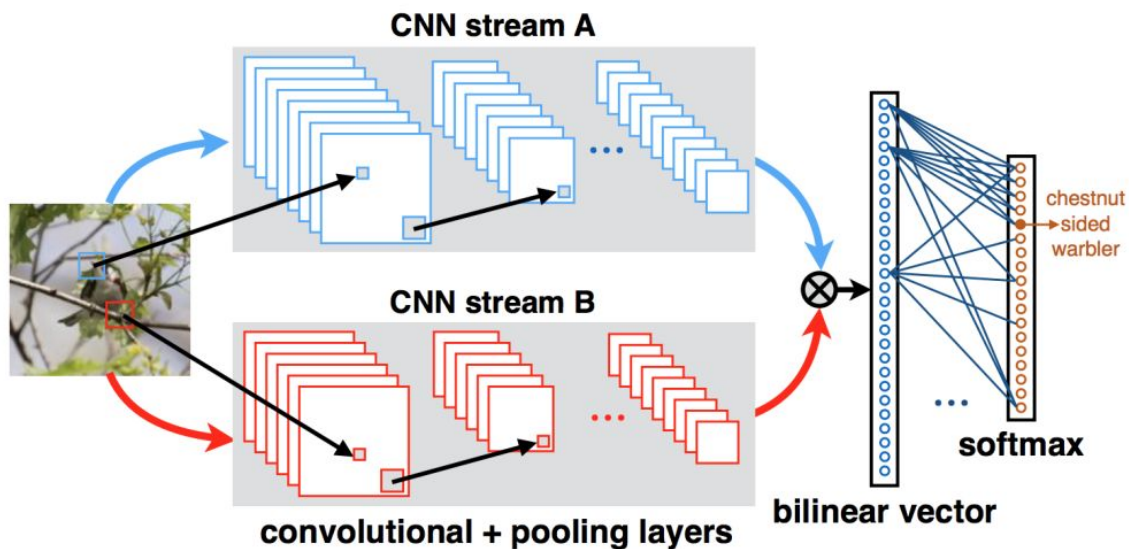


Figure 1. A **bilinear CNN model for image classification**. At test time an image is passed through two CNNs, A and B, and their outputs are multiplied using outer product at each location of the image and pooled to obtain the bilinear vector. This is passed through a classification layer to obtain predictions.

Mohammad Moghimi, Serge J. Belongie, Mohammad J. Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. BMVC 2016.

Moghimi等人提出BoostCNN，利用多个双线性CNN的boosting集成来提升性能，通过最小二乘目标函数，学习boosting权重。然而这会使得训练变慢两个量级。

Algorithm 1 BoostCNN

Input: number of classes M , number of boosting, iterations N_b , shrinkage parameter v , and dataset $\mathcal{D} = \{(x_1, z_1), \dots, (x_n, z_n)\}$ where $z_i \in \{1 \dots M\}$ is label of example x_i .

Init: set $f(x) = \mathbf{0} \in \mathbb{R}^M$.

for $t = 1$ to N_b **do**

 compute $w(x_i)$ for all x_i , using (5).

 train a network $g^*(x)$ to optimize (11).

 find the optimal coefficient, α^* , using (7).

 update $f(x) = f(x) + v\alpha^*g^*(x)$.

end for

Output: predictor $f(x)$

Tsung-Yu Lin and Subhransu Maji. Improved bilinear pooling with CNNs. BMVC 2017.

Lin和Maji探索了对格拉姆矩阵不同的规范化方案，并发现对格拉姆矩阵进行0.5矩阵幂规范化压缩格拉姆矩阵特征值的动态范围

$$G \leftarrow G^{\frac{1}{2}}$$

后结合逐元素平方开根和

规范化可以得到2-3 %的性能提升。此外，由于GPU上没有SVD的高效实现，Lin和Maji使用牛顿迭代法的变体计算 $G^{1/2}$ ，并通过解李雅普诺夫方程（Lyapunov equation）来估计 $G^{1/2}$ 的梯度进行训练。

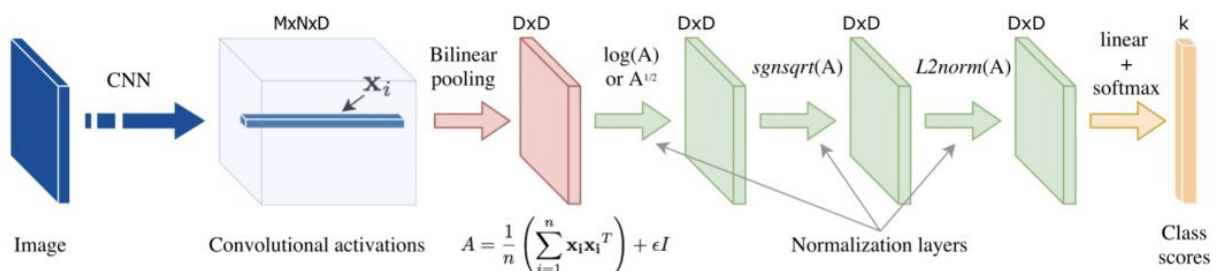


Figure 1: Improved B-CNN architecture with a $\log(A)$ or $A^{1/2}$, signed square-root, and L2 normalization layers added after the bilinear pooling of CNN activations.

Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? ICCV 2017: 2089-2097.

Li等人提出MPN-COV，其对深度描述向量的协方差矩阵 Σ 进行0.5矩阵幂规范化

$$\Sigma \leftarrow \Sigma^{\frac{1}{2}}$$

以得到双线性汇合特征。

Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. CVPR 2018: 947-955.

由于在GPU上没有特征值分解和SVD的高效实现，相比Lin和Maji在反向解李雅普诺夫方程时仍需进行舒尔分解（Schur decomposition）或特征值分解，Li等人前向和反向过程都基于牛顿迭代法，来对矩阵进行0.5矩阵幂规范化。

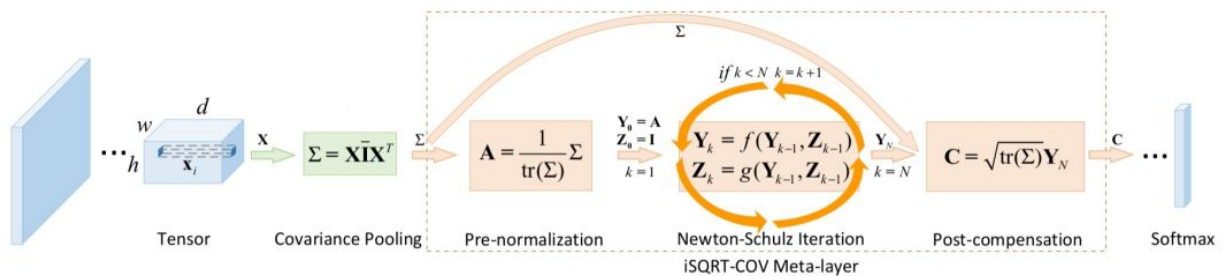


Figure 1. Proposed iterative matrix square root normalization of covariance pooling (iSQRT-COV) network. After the last convolution layer, we perform second-order pooling by estimating a covariance matrix. We design a meta-layer with loop-embedded directed graph structure for computing approximate square root of covariance matrix. The meta-layer consists of three nonlinear structured layers, performing pre-normalization, coupled Newton-Schulz iteration and post-compensation, respectively. See Sec. 3 for notations and details.

Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. ECCV 2018: 365-380.

Wei等人认为，格拉姆矩阵会受视觉爆发（visual burstiness）现象的影响。视觉爆发是指一个视觉元素在同一个图片中多次出现，这会影响其他视觉元素的作用。关于视觉爆发的更多信息请参阅Herve Jegou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. CVPR 2009: 1169-1176.

事实上，由于格拉姆矩阵 $G=(1/HW)XX^T$ 的条件数是 X 的平方，因此格拉姆矩阵会比描述矩阵更不稳定。因此，通过0.5矩阵幂可以使得格拉姆矩阵的条件数等于描述矩阵的条件数，稳定训练过程。

Wei等人提出格拉斯曼（Grassmann）/子空间汇合，令X的SVD分解为 $X=U\Sigma V^T$ ，记矩阵U的前k列为 $U_k \in \mathbb{R}^{D \times K}$ ，该汇合结束输出

$$\tilde{G} = \frac{1}{HW} U_k U_k^T$$

这个结果不受X的条件数的影响。相比经典双线性汇合结果

$$G = \frac{1}{HW} U \Sigma^2 U^T$$

格拉斯曼/子空间汇将格拉姆矩阵的前k个奇异值/特征值置1，其他奇异值/特征值置0。

考虑一张训练图像和一张测试图像对应的格拉斯曼/子空间汇合结果

$$\tilde{G} \text{ 和 } \tilde{G}'$$

$$\begin{aligned} \frac{1}{2} \|\tilde{G} - \tilde{G}'\|_F^2 &= \frac{1}{2} \text{tr} (\tilde{G} - \tilde{G}')^T (\tilde{G} - \tilde{G}') \\ &= \frac{1}{2} \text{tr} \tilde{G} \tilde{G} + \frac{1}{2} \text{tr} \tilde{G}' \tilde{G}' - \text{tr} \tilde{G} \tilde{G}' \\ &= \frac{1}{2} \text{tr} U_k U_k^T U_k U_k^T + \frac{1}{2} \text{tr} U'_k U_k'^T U'_k U_k'^T - \text{tr} U_k U_k^T U'_k U_k'^T \\ &= \frac{1}{2} \text{tr} I + \frac{1}{2} \text{tr} I - \text{tr} U_k U_k^T U'_k U_k'^T \\ &= k - \|U_k'^T U_k\|_F^2. \end{aligned}$$

因此，下一层线性分类器的输出是

$$y = \|\mathbf{W}^T U_k\|_F^2$$

此外，在格拉斯曼/子空间汇合前Wei等人先对描述向量进行降维。Wei等人认为常用的使用1X1卷积核P进行降维会使得降维结果的各通道的多样性降低，因此使用P的奇异向量来降维。

Tsung-Yu Lin, Subhransu Maji, and Piotr Koniusz. Second-Order democratic aggregation. ECCV 2018: 639-656.

Lin等人使用民主聚合来使不同深度描述向量的贡献接近。

Qilong Wang, Peihua Li, and Lei Zhang. G2DeNet: Global Gaussian distribution embedding network and its application to visual recognition. CVPR 2017: 6507-6516.

Wang等人提出G2DeNet，同时利用了一阶和二阶信息

并对其进行0.5矩阵幂规范化。

Mengran Gou, Fei Xiong, Octavia I. Camps, and Mario Sznajder. MoNet: Moments embedding network. CVPR 2018: 3175-3183.

Gou等人对描述矩阵X进行增广

$$\tilde{\mathbf{X}} := \frac{1}{\sqrt{HW}} \begin{bmatrix} \mathbf{X} \\ \mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{(D+1) \times (HW)}$$

这样，通过一次矩阵乘法，可以得到

$$\tilde{\mathbf{G}} := \tilde{\mathbf{X}} \tilde{\mathbf{X}}^\top = \begin{bmatrix} \mathbf{G} & \boldsymbol{\mu}^\top \\ \boldsymbol{\mu} & 1 \end{bmatrix} \in \mathbb{R}^{(D+1) \times (D+1)}$$

其同时包含了一阶和二阶信息。另外，利用tensor sketch，可以设计精简双线性汇合操作。

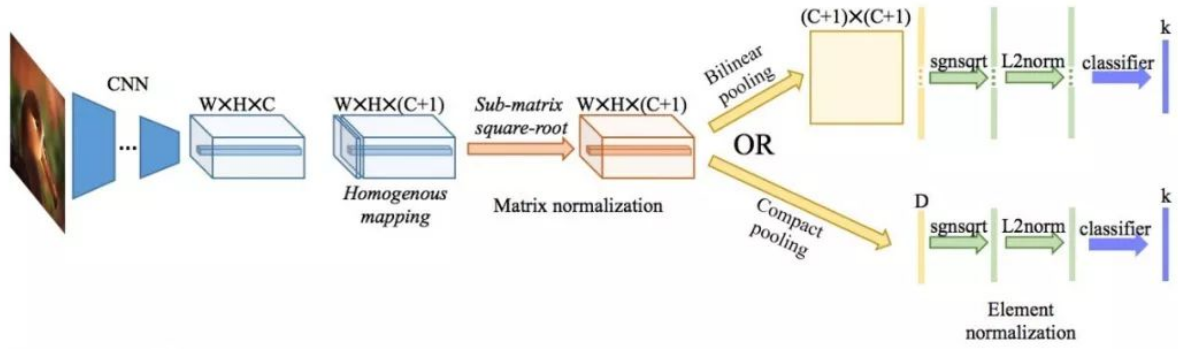


Figure 1. Architecture of the proposed moments-based network **MoNet**. With the proposed sub-matrix square-root layer, it is possible to perform matrix normalization before bilinear pooling or further apply compact pooling to reduce the dimensionality dramatically without undermining performance.

Marcel Simon, Yang Gao, Trevor Darrell, Joachim Denzler, and Erik Rodner.

Generalized orderless pooling performs implicit salient matching. ICCV 2017: 4970-4979.

Simon等人提出 α -汇合，统一表示平均汇合和双线性汇合。 α -汇合形式化为

$$\tilde{G} := \frac{1}{HW} \sum_{i=1}^{HW} (\text{sign } \mathbf{x}_i \odot |\mathbf{x}_i|^{\alpha-1}) \mathbf{x}_i^\top \in \mathbb{R}^{D \times D}$$

其中 α 是一个可学习的参数， $|\mathbf{x}|^{\alpha-1}$ 表示对 \mathbf{x} 逐元素取绝对值并逐元素取幂。当深度描述向量每一项都非负时（这是通常的情况，因为我们使用ReLU层的特征）， α -汇合可以简化为

$$\tilde{G} := \frac{1}{HW} \sum_{i=1}^{HW} \mathbf{x}_i^{\alpha-1} \mathbf{x}_i^\top \in \mathbb{R}^{D \times D}$$

当 $\alpha=1$ 时，代表平均汇合

$$\tilde{G} = \frac{1}{HW} \sum_{i=1}^{HW} \mathbf{1} \mathbf{x}_i^\top = \mathbf{1} \mu^\top$$

当 $\alpha=2$ 时，代表双线性汇合

$$\tilde{\mathbf{G}} = \frac{1}{HW} \sum_{i=1}^{HW} \mathbf{x}_i \mathbf{x}_i^\top$$

此外，为了对深度卷积神经网络学到特征进行理解，考虑一张训练图像和一张测试图像对应的格拉姆矩阵

$$\tilde{\mathbf{G}} \text{ 和 } \tilde{\mathbf{G}}'$$

$$(\text{vec } \tilde{\mathbf{G}})^\top (\text{vec } \tilde{\mathbf{G}}') = \text{tr } \tilde{\mathbf{G}}^\top \tilde{\mathbf{G}}' = \frac{1}{H^2 W^2} \text{tr} \left(\sum_{i=1}^{HW} \mathbf{x}_i \mathbf{x}_i^\top \right)^\top \left(\sum_{j=1}^{HW} \mathbf{x}_j \mathbf{x}_j^\top \right) = \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} \mathbf{x}_i^\top \mathbf{x}_j (\mathbf{x}_i^{\alpha-1})^\top (\mathbf{x}_j^{\alpha-1})$$

其中 χ 是对应于

$$\tilde{\mathbf{G}}$$

的描述向量， χ 是对应于

$$\tilde{\mathbf{G}}'$$

的描述向量。如果后续使用一层全连接层进行分类，这相当于使用了线性核SVM。根据表示定理，测试图像的分类分数是和所有训练图像内积的线性组合

$$\sum_{n=1}^N \beta_n (\text{vec } \tilde{\mathbf{G}}_n)^\top (\text{vec } \tilde{\mathbf{G}}') = \frac{1}{H^2 W^2} \sum_{n=1}^N \sum_{i=1}^{HW} \sum_{j=1}^{HW} \beta_n \mathbf{x}_{ni}^\top \mathbf{x}_j (\mathbf{x}_{ni}^{\alpha-1})^\top (\mathbf{x}_j^{\alpha-1})$$

其中N代表训练样例总数。令

$$\gamma_{nij} := \beta_n \mathbf{x}_{ni}^\top \mathbf{x}_j (\mathbf{x}_{ni}^{\alpha-1})^\top (\mathbf{x}_j^{\alpha-1})$$

，其表示了测试图像的第j个深度描述向量和第n张训练图像的第i个深度描述向量的匹配程度，也就是对应图像区域的匹配程度。通过对 γ_{nij} 进行可视化，我们可以观察哪些区域对于细粒度分类最有帮助。

3. 精简双线性汇合

由于格拉姆矩阵非常高维，有很多工作致力于设计精简双线性汇合，本节分别予以简要介绍。

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. ICCV 2015, pages: 1449-1457.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear convolutional neural networks for fine-grained visual recognition. TPAMI 2018, 40(6): 1309-1322.

Lin等人将双线性汇合输入的一个分支先使用1*1的卷积进行降维，例如将512维降维到64维。以VGG-16为例，最终格拉姆矩阵将由 $R^{512 \times 512}$ 降维到 $R^{512 \times 64}$ 。在实际应用中，用PCA对这个1X1卷积的参数进行初始化。

Yang Gao, Oscar Beijbom, and Ning Zhang, and Trevor Darrell. Compact bilinear pooling. CVPR 2016: 317-326

考虑一张训练图像和一张测试图像对应的格拉姆矩阵 G 和 G' ，

$$(\text{vec } \mathbf{G})^\top (\text{vec } \mathbf{G}') = \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} (\mathbf{x}_i^\top \mathbf{x}_j)^2$$

Gao等人使用深度描述向量的低维投影的内积近似二次多项式核

$$(\text{vec } \mathbf{G})^\top (\text{vec } \mathbf{G}') \approx \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

实际应用中，Gao等人使用了random Maclaurin和tensor sketch两种近似，可以在达到和标准双线性汇合相似的性能，但参数量减少了90%。

Algorithm 1 Random Maclaurin Projection

Input: $x \in \mathbb{R}^c$

Output: feature map $\phi_{RM}(x) \in \mathbb{R}^d$, such that $\langle \phi_{RM}(x), \phi_{RM}(y) \rangle \approx \langle x, y \rangle^2$

1. Generate random but fixed $W_1, W_2 \in \mathbb{R}^{d \times c}$, where each entry is either $+1$ or -1 with equal probability.
 2. Let $\phi_{RM}(x) \equiv \frac{1}{\sqrt{d}}(W_1 x) \circ (W_2 x)$, where \circ denotes element-wise multiplication.
-

Algorithm 2 Tensor Sketch Projection

Input: $x \in \mathbb{R}^c$

Output: feature map $\phi_{TS}(x) \in \mathbb{R}^d$, such that $\langle \phi_{TS}(x), \phi_{TS}(y) \rangle \approx \langle x, y \rangle^2$

1. Generate random but fixed $h_k \in \mathbb{N}^c$ and $s_k \in \{+1, -1\}^c$ where $h_k(i)$ is uniformly drawn from $\{1, 2, \dots, d\}$, $s_k(i)$ is uniformly drawn from $\{+1, -1\}$, and $k = 1, 2$.
 2. Next, define sketch function $\Psi(x, h, s) = \{(Qx)_1, \dots, (Qx)_d\}$, where $(Qx)_j = \sum_{t:h(t)=j} s(t)x_t$
 3. Finally, define $\phi_{TS}(x) \equiv \text{FFT}^{-1}(\text{FFT}(\Psi(x, h_1, s_1)) \circ \text{FFT}(\Psi(x, h_2, s_2)))$, where the \circ denotes element-wise multiplication.
-

关于利用FFT进行快速多项式乘法的部分，已超出本文范围。感兴趣的读者可参阅 Selçuk Baktir and Berk Sunar. Achieving efficient polynomial multiplication in fermat fields using the fast Fourier transform. ACM Southeast Regional Conference 2006: 549-554.

Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge J. Belongie.
Kernel pooling for convolutional neural networks. CVPR 2017: 3049-3058.

Cui等人进一步利用核近似，借助于tensor sketch捕获更高阶的特征交互，提出核汇合。

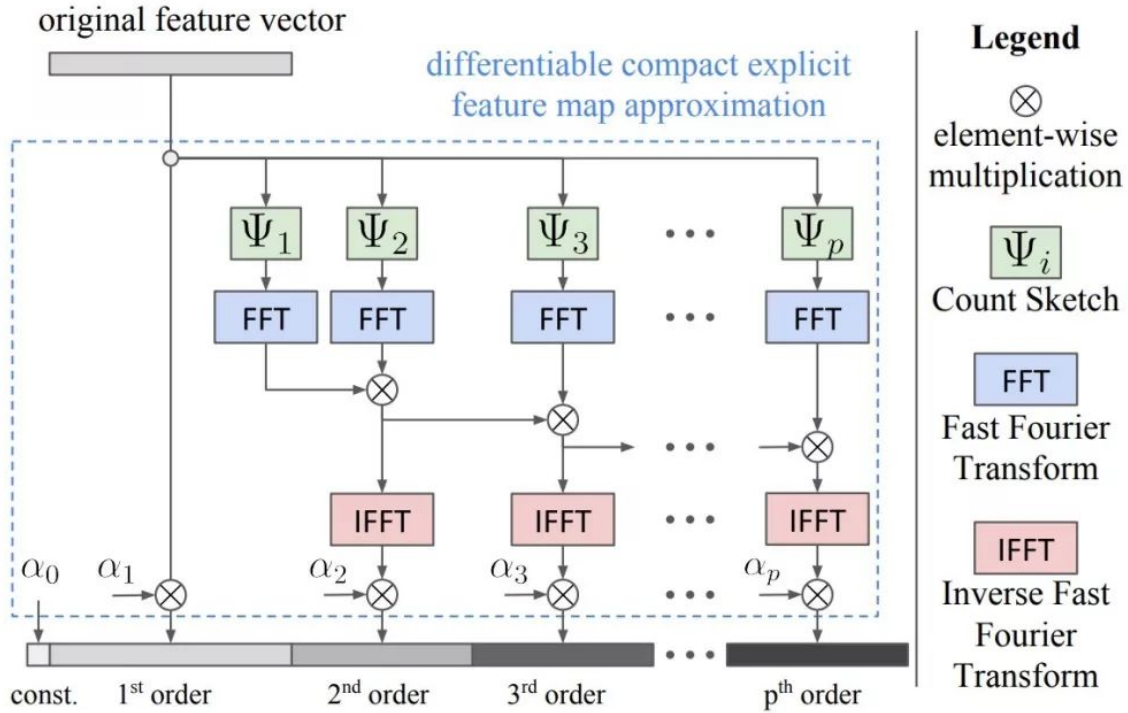


Figure 1. The proposed Kernel Pooling method. For a feature vector (i.e., the activation at a spatial location on the feature map, in the case of a CNN), we use Count Sketch [6] to generate a compact explicit feature map up to p^{th} order. After applying kernel pooling, the inner product between two features can capture high order feature interactions as in Eqn. 1. This makes the subsequent linear classifier highly discriminative. The proposed kernel pooling scheme is end-to-end trainable and the composition of the kernel can be learned through the update of coefficients $\{\alpha_i\}_{i=0}^p$. The vanilla compact bilinear pooling [11, 10] only use the 2^{nd} order information as the feature vector.

Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. NIPS 2018, accepted.

Yue等人将非局部 (non-local) 网络 (参见下文) 应用到细粒度识别中，并对核函数泰勒展开进行近似。

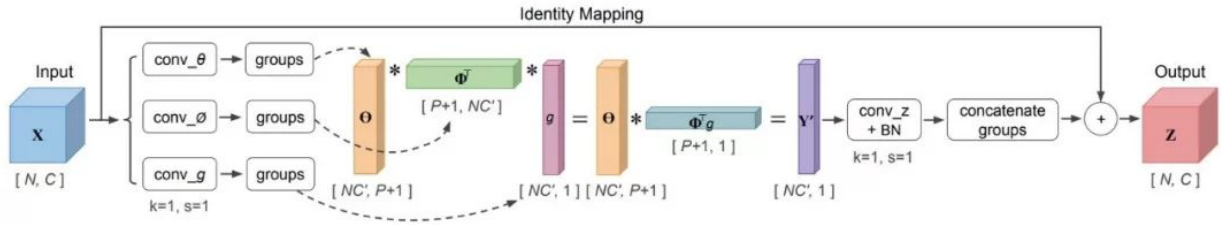


Figure 2: Grouped compact generalized non-local (CGNL) module. The feature maps are shown with the shape of their tensors, e.g., $[C, N]$, where $N = THW$ or $N = HW$. The feature maps will be divided along channels into multiple groups after three conv layers whose kernel size and stride both equals 1 ($k = 1, s = 1$). The channels dimension is grouped into $C' = C/G$, where G is a group number. The compact representations for generalized non-local module are build within each group. P indicates the order of Taylor expansion for kernel functions.

Shu Kong and Charless C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. CVPR 2017: 7025-7034.

经典的双线性汇合算法在提取双线性特征后使用线性分类器（例如使用一层全连接层或者使用线性SVM）

$$y := (\text{vec } \mathbf{W})^\top (\text{vec } \mathbf{G}) + b = \frac{1}{HW} (\text{vec } \mathbf{W})^\top (\text{vec } \mathbf{X} \mathbf{X}^\top) + b$$

根据表示定理，最优的参数矩阵是特征的线性组合

$$\mathbf{W} = \sum_{n=1}^N \beta_n \mathbf{X} \mathbf{X}^\top$$

因此，最优的参数矩阵是对阵矩阵，我们可以对其做谱分解，并根据特征值的正负分成两部分

如果我们把所有大于0的特征值和特征向量汇总到矩阵

$$\mathbf{W} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top = \sum_{d=1}^D \lambda_d \mathbf{q}_d \mathbf{q}_d^\top = \sum_{d: \lambda_d > 0} \lambda_d \mathbf{q}_d \mathbf{q}_d^\top - \sum_{d: \lambda_d < 0} |\lambda_d| \mathbf{q}_d \mathbf{q}_d^\top$$

以及所有小于0的特征值和特征向量汇总到矩阵

$$\mathbf{U}_+ := [\sqrt{|\lambda_{1+}|} \mathbf{q}_{1+} \ \sqrt{|\lambda_{2+}|} \mathbf{q}_{2+} \ \cdots \ \sqrt{|\lambda_{\frac{r}{2}+}|} \mathbf{q}_{\frac{r}{2}+}] \in \mathbb{R}^{D \times \frac{r}{2}}$$

那么,

$$\mathbf{W} = \mathbf{U}_+ \mathbf{U}_+^\top - \mathbf{U}_- \mathbf{U}_-^\top$$

即, 我们可以用参数矩阵 $\mathbf{U} := [\mathbf{U}_+ \mathbf{U}_-] \in \mathbb{R}^{D \times r}$ 近似原来参数矩阵 $\mathbf{W} \in \mathbb{R}^{D \times D}$, 其中 r 是一个超参数。这里需要假设参数矩阵大于0和小于0的特征值个数都是 $r/2$ 。

上述低秩近似还有另外一个好处。在计算图像属于各个类别的分数时

$$\begin{aligned} y &= \frac{1}{HW} (\text{vec } \mathbf{W})^\top (\text{vec } \mathbf{X} \mathbf{X}^\top) + b \\ &= \frac{1}{HW} \text{tr } \mathbf{W} \mathbf{X} \mathbf{X}^\top + b \\ &= \frac{1}{HW} \text{tr } (\mathbf{U}_+ \mathbf{U}_+^\top - \mathbf{U}_- \mathbf{U}_-^\top) \mathbf{X} \mathbf{X}^\top + b \\ &= \frac{1}{HW} \text{tr } \mathbf{U}_+ \mathbf{U}_+^\top \mathbf{X} \mathbf{X}^\top - \frac{1}{HW} \text{tr } \mathbf{U}_- \mathbf{U}_-^\top \mathbf{X} \mathbf{X}^\top + b \\ &= \frac{1}{HW} \text{tr } (\mathbf{U}_+^\top \mathbf{X})^\top (\mathbf{U}_+^\top \mathbf{X}) - \frac{1}{HW} \text{tr } (\mathbf{U}_-^\top \mathbf{X})^\top (\mathbf{U}_-^\top \mathbf{X}) + b \\ &= \frac{1}{HW} \|\mathbf{U}_+^\top \mathbf{X}\|_F^2 - \frac{1}{HW} \|\mathbf{U}_-^\top \mathbf{X}\|_F^2 + b. \end{aligned}$$

从上式可以发现, 我们不再需要显式的计算双线性汇合结果。

为了进一步降低参数矩阵的计算量, LRBP对参数矩阵做了第二次近似。由于细粒度识别问题中各类有公共结构, 因此, LRBP将对应各类别的参数矩阵近似为 $\mathbf{W}_k = \mathbf{P} \mathbf{V}_k$, 其中 $\mathbf{P} \in \mathbb{R}^{D \times m}$, m 是另一个超参数。不同类别的参数矩阵 \mathbf{P} 是共享的, 实际中, \mathbf{P} 和 \mathbf{V}_k 都由 1×1 卷积实现。 \mathbf{P} 由PCA投影矩阵初始化, 而 \mathbf{V}_k 采用随机初始化。

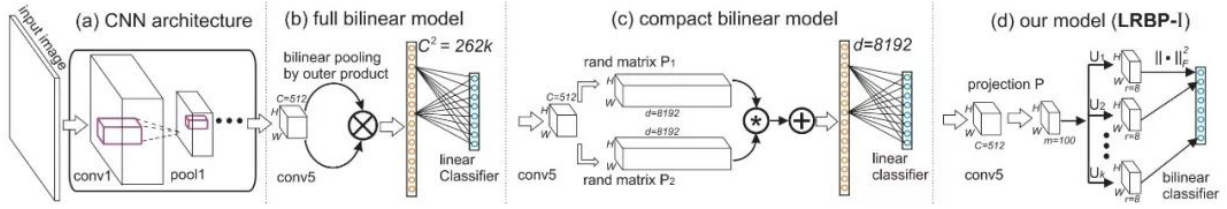


Figure 1: We explore models that perform classification using second order statistics of a convolutional feature map (a) as input (e.g., VGG16 layer *conv5_3*). Architecture of (b) full bilinear model [20], (c) recently proposed compact bilinear model [8], and (d) our proposed low-rank bilinear pooling model (LRBP). Our model captures second order statistics without explicitly computing the pooled bilinear feature, instead using a bilinear classifier that uses the Frobenius norm as the classification score. A variant of our architecture that exploits co-decomposition and computes low-dimensional bilinear features is sketched in Figure 4.

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. ICCV 2017: 2098-2106.

Li等人通过对参数矩阵进行低秩近拟来完成图像分类任务，同时其也包含了一阶信息。Li等人所提出的FBN结构可以用于所有的卷积和全连接层。

Sijia Cai, Wangmeng Zuo, and Lei Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. ICCV 2017: 511-520.

Cai等人捕获了更高阶的特征交互，并对参数进行秩1近似。

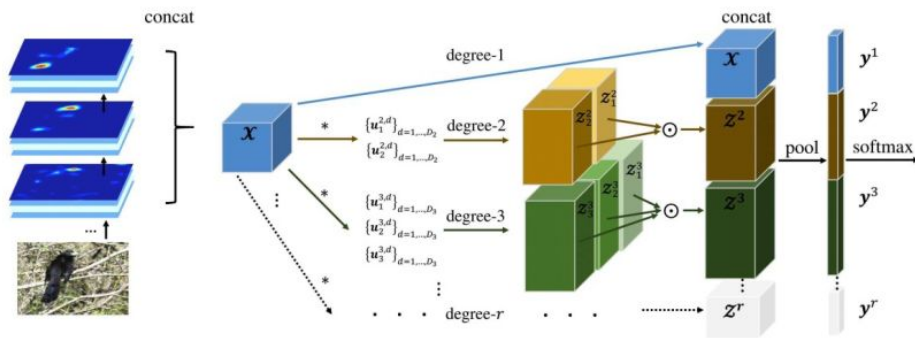


Figure 2. Illustration of our integration framework. The convolutional activation maps are concatenated as $\mathcal{X} = \text{concat}(\mathcal{X}^1, \dots, \mathcal{X}^L)$ and fed into different branches. For r -th branch ($r \geq 2$), the degree- r polynomial module consisting of r groups of 1×1 convolutional filters is deployed to obtain r sets of feature maps $\{\mathcal{Z}_s^r\}_{s=1, \dots, r}$. Then $\{\mathcal{Z}_s^r\}_{s=1, \dots, r}$ are integrated as \mathcal{Z}^r by applying element-wise product \odot . At last, \mathcal{X} and all \mathcal{Z}^r s are concatenated as the degree- r polynomial features, following by sum pooling layer to obtain the pooled representation $\mathbf{y} = \text{concat}(\mathbf{y}^1, \dots, \mathbf{y}^L)$ with the dimension of $\sum_{r=1}^R D_r$ (D_1 denotes the channel number of \mathcal{X}), and softmax layer.

Kaicheng Yu and Mathieu Salzmann. Statistically-motivated second-order pooling. ECCV 2018: 621-637.

Yu和Salzmann对参数矩阵进行低秩近似，并给双线性汇合的每一步一个概率上的解释。这个操作可以被等价于对深度描述向量进行1X1卷积后在

汇合。

Algorithm 1: SMSO Pooling

Input : Data matrix $\hat{\mathbf{X}} \in \mathbb{R}^{n \times c}$,
 Projection dimension p .
Output: $\mathbf{z}'' \in \mathbb{R}^p$, where $\mathbf{z}_j'' \sim \mathcal{N}(\beta_j, \gamma_j)$.
 1 Initialization: $\forall j \in [1, p], \forall l \in [1, c], \beta_j = 0, \gamma_j = 1$,
 $\mathbf{w}_j \in \mathbb{R}^c, [\mathbf{w}_j]_l \sim \mathcal{N}(0, \sqrt{\frac{2}{c+p}})$ [13];
 2 $\mathbf{X} \leftarrow \text{BatchNorm}(\hat{\mathbf{X}})$, where $\mathbf{x}_i \sim \mathcal{N}_c(\boldsymbol{\mu}, \boldsymbol{\Sigma})$;
 3 $\mathbf{Y} \leftarrow \text{Cov}(\mathbf{X}) = \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$;
 4 **for** $j \in [1, p]$ **do**
 5 $\mathbf{z}_j \leftarrow \text{PV}(\mathbf{Y}, j) = \mathbf{w}_j^T \mathbf{Y} \mathbf{w}_j$;
 6 $\alpha \leftarrow 2/(\mathbf{w}_j^T \boldsymbol{\Sigma} \mathbf{w}_j)$;
 7 $\mathbf{z}_j' \leftarrow \text{ScaledSqrt}(\mathbf{z}_j) = \sqrt{\alpha \mathbf{z}_j} - \sqrt{2n-1}$;
 8 $\mathbf{z}_j'' \leftarrow \text{Norm}(\mathbf{z}_j') = \beta_j + \gamma_j \mathbf{z}_j'$;
 9 **end**

Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. ECCV 2018: 595-610.

类似于MLB（见下文），对参数矩阵进行低秩近似

$$\mathbf{y} = \mathbf{P}^\top (\mathbf{U}^\top \mathbf{x} \odot \mathbf{V}^\top \mathbf{z}) + b$$

并使用不同层的深度描述向量作为 \mathbf{x} 和 \mathbf{z} 。

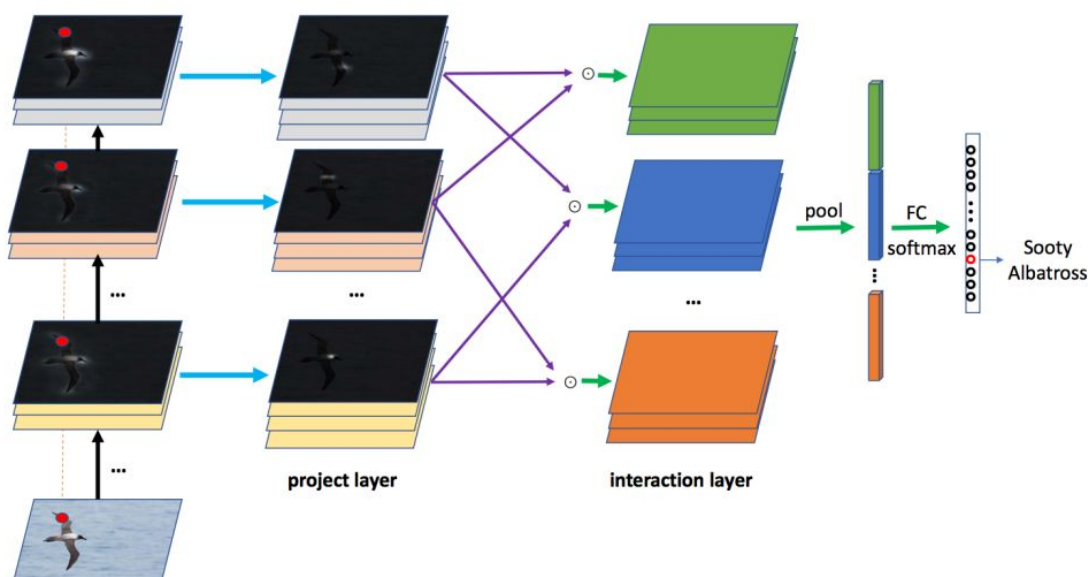


Fig. 1. Illustration of our Hierarchical Bilinear Pooling (HBP) network architecture for fine-grained recognition. The bottom image is the input, and above it are the feature maps of different layers in the CNN. First the features from different layers are expanded into a high-dimensional space via independent linear mapping to capture attributes of different object parts and then integrated by element-wise multiplication to model the inter-layer interaction of part attributes. After that sum pooling is performed to squeeze the high-dimensional features into compact ones. Note that we obtain the visual activation maps above by computing the response of sum-pooled feature vector on every single spatial location.

4. 双线性汇合的其他应用

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. Image style transfer using convolutional neural networks. CVPR 2016: 2414-2423.

风格迁移的基本思路是使优化目标包括两项，使生成图像的内容接近原始图像内容，及使生成图像风格接近给定风格。风格通过格拉姆矩阵体现，而内容则直接通过神经元激活值体现。

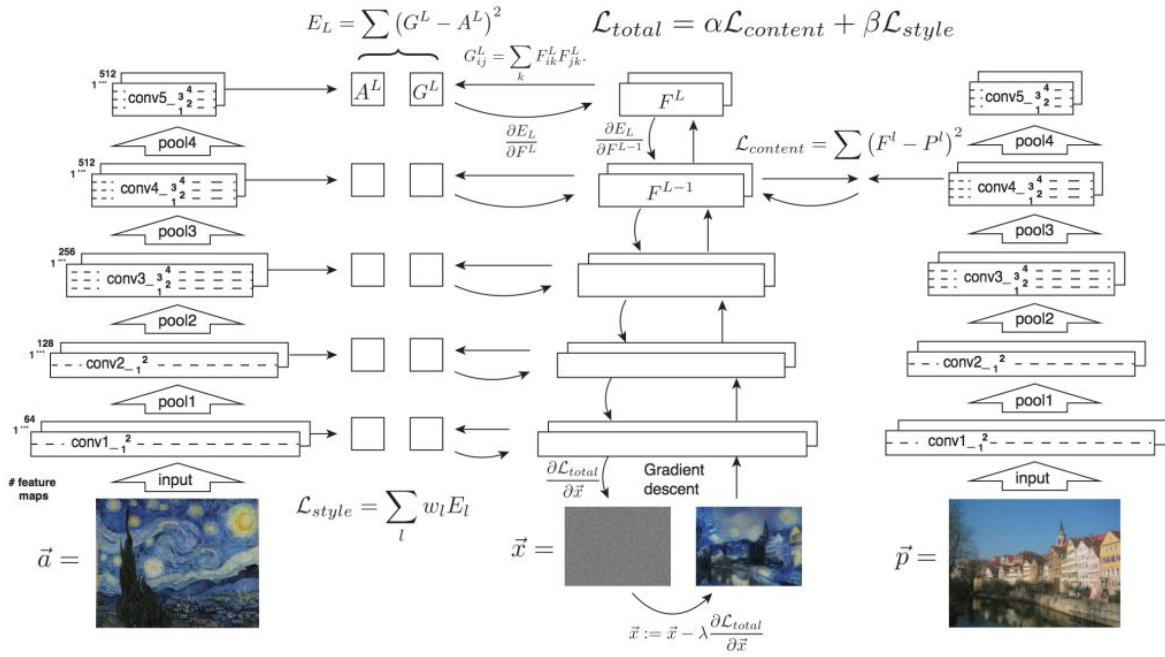


Figure 2. Style transfer algorithm. First content and style features are extracted and stored. The style image \vec{a} is passed through the network and its style representation A^l on all layers included are computed and stored (left). The content image \vec{p} is passed through the network and the content representation P^l in one layer is stored (right). Then a random white noise image \vec{x} is passed through the network and its style features G^l and content features F^l are computed. On each layer included in the style representation, the element-wise mean squared difference between G^l and A^l is computed to give the style loss \mathcal{L}_{style} (left). Also the mean squared difference between F^l and P^l is computed to give the content loss $\mathcal{L}_{content}$ (right). The total loss \mathcal{L}_{total} is then a linear combination between the content and the style loss. Its derivative with respect to the pixel values can be computed using error back-propagation (middle). This gradient is used to iteratively update the image \vec{x} until it simultaneously matches the style features of the style image \vec{a} and the content features of the content image \vec{p} (middle, bottom).

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *IJCAI 2017*: 2230-2236.

Li等人发现，考虑一张训练图像和一张测试图像对应的格拉姆矩阵 G 和 G' ，风格损失项

$$\begin{aligned} \|G - G'\|_F^2 &= \text{tr}(G - G')^\top (G - G') \\ &= \text{tr} G^\top G + \text{tr} G'^\top G' - 2 \text{tr} G^\top G' \\ &= \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} (x_i^\top x_j)^2 + \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} (x'_i{}^\top x'_j)^2 - \frac{1}{H^2 W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} (x_i^\top x'_j)^2 \end{aligned}$$

这里利用了上文Compact bilinear pooling的推导结果。这对应了深度描述向量之间的二阶多项式MMD距离。最小化风格损失就是在最小化这两者的特征分布。

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *EMNLP 2016*: 457-468.

Fukui等人提出MCB，对图像和文本特征进行精简双线性汇合。

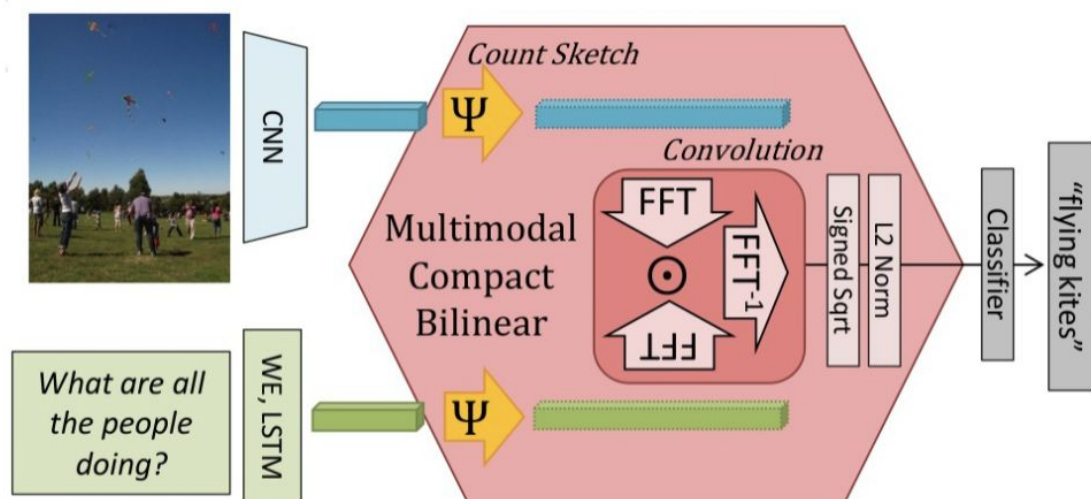


Figure 1: Multimodal Compact Bilinear Pooling for visual question answering.

Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, JungWoo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. ICLR 2017.

Kim等人提出MLB，对参数矩阵进行低秩近似。假设图像描述向量是 x ，文本描述向量是 z ，那么它们的双线性汇合的格拉姆矩阵是

$$G = xz^T$$

在提取双线性特征后使用线性分类器

$$y := (\text{vec } W)^T (\text{vec } G) + b = (\text{vec } W)^T (\text{vec } xz^T) + b = \text{tr } W^T xz^T + b = x^T Wz + b$$

对参数矩阵进行低秩近似 $W=UV^T$ ，这样

$$y = x^T UV^T z + b = (U^T x)^T (V^T z) + b = \mathbf{1}^T (U^T x \odot V^T z) + b$$

对整个分数向量，使用一个投影矩阵，并增加激活函数

$$y = P^T \text{sigm}(U^T x \odot V^T z) + b$$

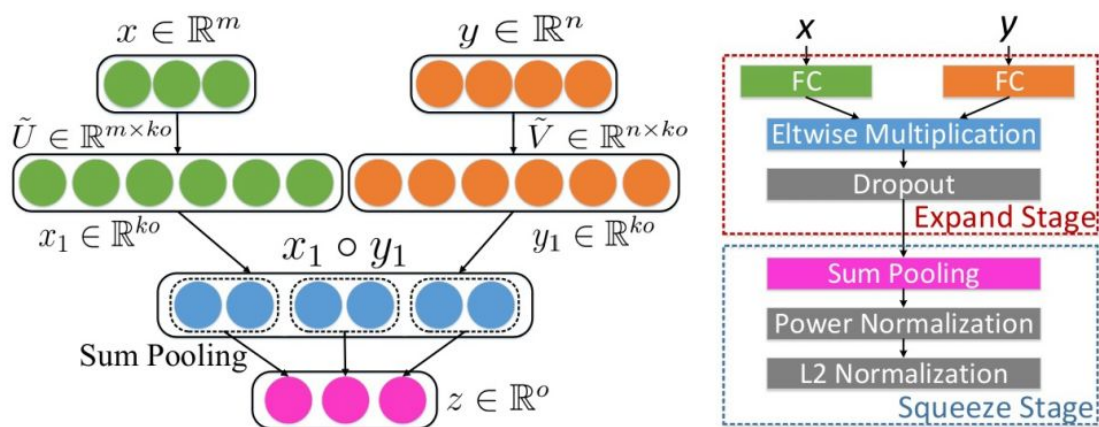
Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. ICCV 2017: 1839-1848.

Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond bilinear: Generalized multi-modal factorized high-order pooling for visual question answering. TNNLS 2018, in press.

Yu等人提出MFB，也是对参数矩阵进行低秩近似。和MLB相比，分数向量 y 的每一项都由

$$y = \mathbf{1}^\top (U^\top x \odot V^\top z) + b$$

计算得到，不进一步进入投影矩阵和激活函数。



(a) Multi-modal Factorized Bilinear Pooling

(b) MFB module

Figure 1. The flowchart of Multi-modal Factorized Bilinear Pooling and completed design of the MFB module.

Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. NIPS 2017: 33-44.

Girdhar和Remanan对参数矩阵做了秩1近似 $W=UVT$ ，即

$$\begin{aligned}
y &= \frac{1}{HW} (\text{vec } \mathbf{W})^\top (\text{vec } \mathbf{X} \mathbf{X}^\top) + b \\
&= \frac{1}{HW} \text{tr } \mathbf{W} \mathbf{X} \mathbf{X}^\top + b \\
&= \frac{1}{HW} \text{tr } \mathbf{u} \mathbf{v}^\top \mathbf{X} \mathbf{X}^\top + b \\
&= \frac{1}{HW} \text{tr } \mathbf{v}^\top \mathbf{X} \mathbf{X}^\top \mathbf{u} + b \\
&= \frac{1}{HW} (\mathbf{v}^\top \mathbf{X})(\mathbf{u}^\top \mathbf{X})^\top + b.
\end{aligned}$$

对整个分数向量， \mathbf{u} 对所有类别共享，不同类别有各自独立的 \mathbf{v} ，这两个分别表示为top-down attention和bottom-up attention。

Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. CVPR 2018: 7794-7803.

非局部 (non-local) 操作可用于捕获长距离依赖

$$y_i := \frac{1}{C(\mathbf{x})} \sum_{j=1}^{HW} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j)$$

其中某一位置 i 的响应 y_i 是所有位置响应的加权平均。当 f 是内积， g 是恒等变换时

$$y_i = \frac{1}{HW} \sum_{j=1}^{HW} \mathbf{x}_i^\top \mathbf{x}_j \mathbf{x}_j = \frac{1}{HW} \sum_{j=1}^{HW} \mathbf{x}_j \mathbf{x}_j^\top \mathbf{x}_i = \mathbf{G} \mathbf{x}_i$$

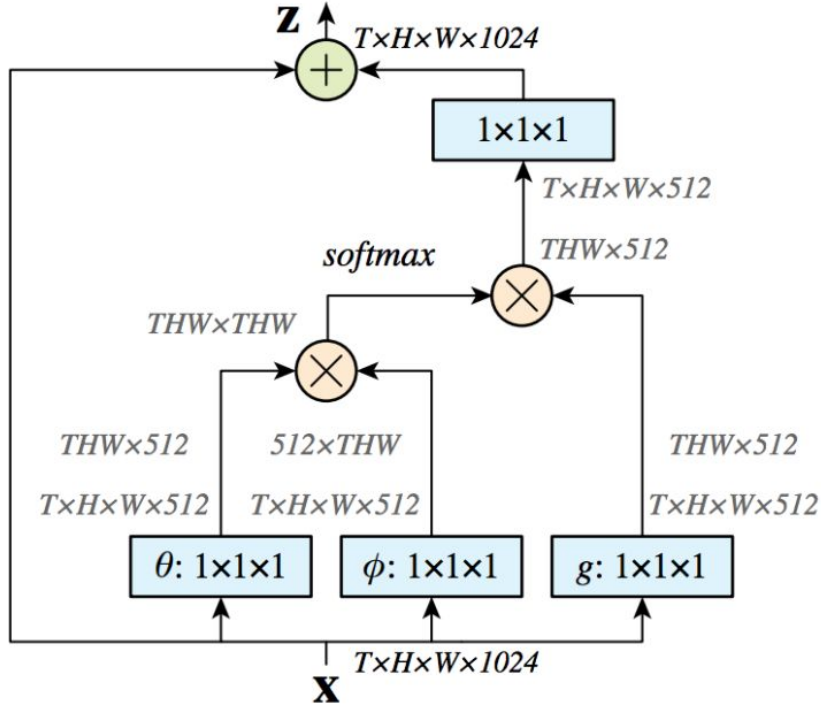


Figure 2. A spacetime **non-local block**. The feature maps are shown as the shape of their tensors, *e.g.*, $T \times H \times W \times 1024$ for 1024 channels (proper reshaping is performed when noted). “ \otimes ” denotes matrix multiplication, and “ \oplus ” denotes element-wise sum. The softmax operation is performed on each row. The blue boxes denote $1 \times 1 \times 1$ convolutions. Here we show the embedded Gaussian version, with a bottleneck of 512 channels. The vanilla Gaussian version can be done by removing θ and ϕ , and the dot-product version can be done by replacing softmax with scaling by $1/N$.

5. 总结

本文介绍了双线性汇合在细粒度图像分类及其他计算机视觉领域的应用。研究方向大致分为两类，设计更好的双线性汇合过程，以及精简双线性汇合。未来可能的研究方向包括以下几个部分

- 双线性汇合结果的规范化。目前发现矩阵幂规范化对提升性能很有帮助，是否还有其他规范化方法，以及能否设计出其在GPU的高效实现？
- 精简双线性汇合。目前研究的一大关注点是对参数矩阵进行低秩近似，是否还有其他精简双线性汇合方法？

- 双线性汇合原理。目前对双线性汇合原理的理解围绕着深度描述向量之间的二阶多项式核展开，是否还有其他对双线性汇合原理的理解方法？

此外，经典双线性汇合官方源代码是基于MatConvNet实现，这里提供一个基于PyTorch的实现版本：<https://github.com/HaoMood/bilinear-cnn>

▲向上滑动查看更多推荐文章

十一月后文章，本公众号不提供PDF版本，请移步www.sigai.cn

📖知识库📖技术文章，获取PDF版本。



科普类

【获取码】SIGAI0413

[机器学习——波澜壮阔四十年](#)

【获取码】SIGAI0620

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第1季从 \$\sqrt{2}\$ 谈起](#)

【获取码】SIGAI0702

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第2季 神经计算的历史背景](#)

【获取码】SIGAI0713

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第3季 神经网络的数学模型](#)

【获取码】SIGAI0815

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第4季 凛冬将至](#)

【获取码】SIGAI1019

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第5季 导数的前生今生](#)

【获取码】SIGAI0829

[AI时代大点兵——国内外知名AI公司2018年最新盘点【完整版】](#)

【获取码】SIGAI1008

[非算法类人工智能从业者须知的十件事](#)

【获取码】SIGAI1102

[永远的金大侠——人工智能的江湖](#)



数学类

【获取码】SIGAI0417

[学好机器学习需要哪些数学知识](#)

【获取码】SIGAI0511

[理解梯度下降法](#)

【获取码】SIGAI0518

[理解凸优化](#)

【获取码】SIGAI0531

[理解牛顿法](#)

【获取码】SIGAI1031

[理解概率密度函数](#)



机器学习类

【获取码】SIGAI0428

[用一张图理解SVM的脉络](#)

【获取码】SIGAI0505

[理解神经网络的激活函数](#)

【获取码】SIGAI0522

[【实验】理解SVM核函数和参数的作用](#)

【获取码】SIGAI0601

[【群话题精华】五月集锦—机器学习和深度学习中一些值得思考的问题](#)

【获取码】SIGAI0602

[大话AdaBoost算法](#)

【获取码】SIGAI0606

[理解主成分分析（PCA）](#)

【获取码】SIGAI0611

[理解决策树](#)

【获取码】SIGAI0613

[用一句话总结常用的机器学习算法](#)

【获取码】SIGAI0618

[理解过拟合](#)

【获取码】SIGAI0627

[k近邻算法](#)

【获取码】SIGAI0704

[机器学习算法地图](#)

【获取码】SIGAI0706

[反向传播算法推导—全连接神经网络](#)

【获取码】SIGAI0720

[流形学习概论](#)

【获取码】SIGAI0725

[随机森林概述](#)

【获取码】SIGAI0711

[怎样成为一名优秀的算法工程师](#)

【获取码】SIGAI0802

[机器学习和深度学习中值得弄清楚的一些问题](#)

【获取码】SIGAI0808

[机器学习和深度学习核心知识点总结--写在校园招聘即将开始前](#)

【获取码】SIGAI0822

[机器学习中的最优化算法总结](#)

【获取码】SIGAI0824

[浓缩就是精华--SIGAI机器学习蓝宝书](#)

【获取码】SIGAI0831

[理解AdaBoost算法](#)

【获取码】SIGAI0903

[深入浅出聚类算法](#)

【获取码】SIGAI0905

[机器学习发展历史回顾](#)

【获取码】SIGAI0914

[理解Logistic回归](#)

【获取码】SIGAI0917

[机器学习中的目标函数总结](#)

【获取码】SIGAI0921

[机器学习与深度学习常见面试题](#)

【获取码】SIGAI1008

[理解生成模型与判别模型](#)

【获取码】SIGAI1024

[机器学习和深度学习常见面试题（下）](#)

【获取码】SIGAI1114

[机器学习在信用评级卡中的应用](#)

深度学习类

【获取码】SIGAI0426

[卷积神经网络为什么能够称霸计算机视觉领域？](#)

【获取码】SIGAI0508

[深度卷积神经网络演化历史及结构改进脉络-40页长文全面解读](#)

【获取码】SIGAI0515

[循环神经网络综述—语音识别与自然语言处理的利器](#)

【获取码】SIGAI0625

[卷积神经网络的压缩与加速](#)

【获取码】SIGAI0709

[生成式对抗网络模型综述](#)

【获取码】SIGAI0711

[怎样成为一名优秀的算法工程师](#)

【获取码】SIGAI0718

[基于深度负相关学习的人群计数方法](#)

【获取码】SIGAI0723

[关于感受野的总结](#)

【获取码】SIGAI0802

[机器学习和深度学习中值得弄清楚的一些问题](#)

【获取码】SIGAI0806

[反向传播算法推导--卷积神经网络](#)

【获取码】SIGAI0808

[机器学习和深度学习核心知识点总结--写在校园招聘即将开始前](#)

【获取码】SIGAI0810

[理解SpatialTransformer Networks](#)

【获取码】SIGAI0822

[机器学习中的最优化算法总结](#)

【获取码】SIGAI0824

[浓缩就是精华--SIGAI机器学习蓝宝书](#)

【获取码】SIGAI0827

[DenseNet详解](#)

【获取码】SIGAI0905

[机器学习发展历史回顾](#)

【获取码】SIGAI0907

[网络表征学习综述](#)

【获取码】SIGAI0917

[机器学习中的目标函数总结](#)

【获取码】SIGAI0921

[机器学习与深度学习常见面试题](#)

【获取码】SIGAI0928

[轻量化神经网络综述](#)

【获取码】SIGAI1015

[三维深度学习中的目标分类与语义分割](#)

【获取码】SIGAI1017

[化秋毫为波澜：运动放大算法\(深度学习版\)](#)

【获取码】SIGAI1022

[深度多目标跟踪算法检测](#)

【获取码】SIGAI1024

[机器学习和深度学习常见面试题（下）](#)

机器视觉类

【获取码】SIGAI0420

[人脸识别算法演化史](#)

【获取码】SIGAI0424

[基于深度学习的目标检测算法综述](#)

【获取码】SIGAI0426

[卷积神经网络为什么能够称霸计算机视觉领域？](#)

【获取码】SIGAI0503

[人脸检测算法综述](#)

【获取码】SIGAI0525

[【SIGAI综述】行人检测算法](#)

【获取码】SIGAI0604

[FlowNet到FlowNet2.0：基于卷积神经网络的光流预测算法](#)

【获取码】SIGAI0608

[人体骨骼关键点检测综述](#)

【获取码】SIGAI0615

[目标检测算法之YOLO](#)

【获取码】SIGAI0622

[场景文本检测——CTPN算法介绍](#)

【获取码】SIGAI0629

[自然场景文本检测识别技术综述](#)

【获取码】SIGAI0716

[人脸检测算法之S3FD](#)

【获取码】SIGAI0727

[基于内容的图像检索技术综述--传统经典方法](#)

【获取码】SIGAI0817

[基于内容的图像检索技术综述--CNN方法](#)

【获取码】SIGAI0910

[视觉多目标跟踪算法综述（上）-附开源代码下载链接整理](#)

【获取码】SIGAI0914

[计算机视觉技术self-attention最新进展](#)

【获取码】SIGAI0919

[人脸识别中的活体检测算法综述](#)

【获取码】SIGAI0924

[浅谈动作识别TSN, TRN, ECO](#)

【获取码】SIGAI0926

[OCR技术简介](#)

【获取码】SIGAI1012

[目标检测最新进展总结与展望](#)

【获取码】SIGAI1029

[目标检测算法中检测框合并策略技术综述](#)

【获取码】SIGAI1112

[图像分割技术介绍](#)

自然语言处理

【获取码】SIGAI0803

[基于深度神经网络的自动问答概述](#)

【获取码】SIGAI0820

[文本表示简介](#)

工业应用类

【获取码】SIGAI0529

[机器学习在自动驾驶中的应用-以百度阿波罗平台为例【上】](#)

[本文为SIGAI原创](#)

[如需转载，欢迎发消息到本订阅号](#)