Efficient Codebook and Factorization for Second-Order Representation Learning

Pierre JACOB[1], David PICARD[1,2], Aymeric HISTACE[1], Edouard KLEIN[3]

1. ETIS, UMR 8051, Université Paris Seine, Université de Cergy-Pontoise, ENSEA, CNRS, F-95000, Cergy, France
2. LIGM, UMR 8049, Ecole des Ponts, UPE, Champs-sur-Marne, France
3. C3N, Pôle Judiciaire de la Gendarmerie Nationale, 5 Boulevard de l'Hautil, 95000 Cergy, France

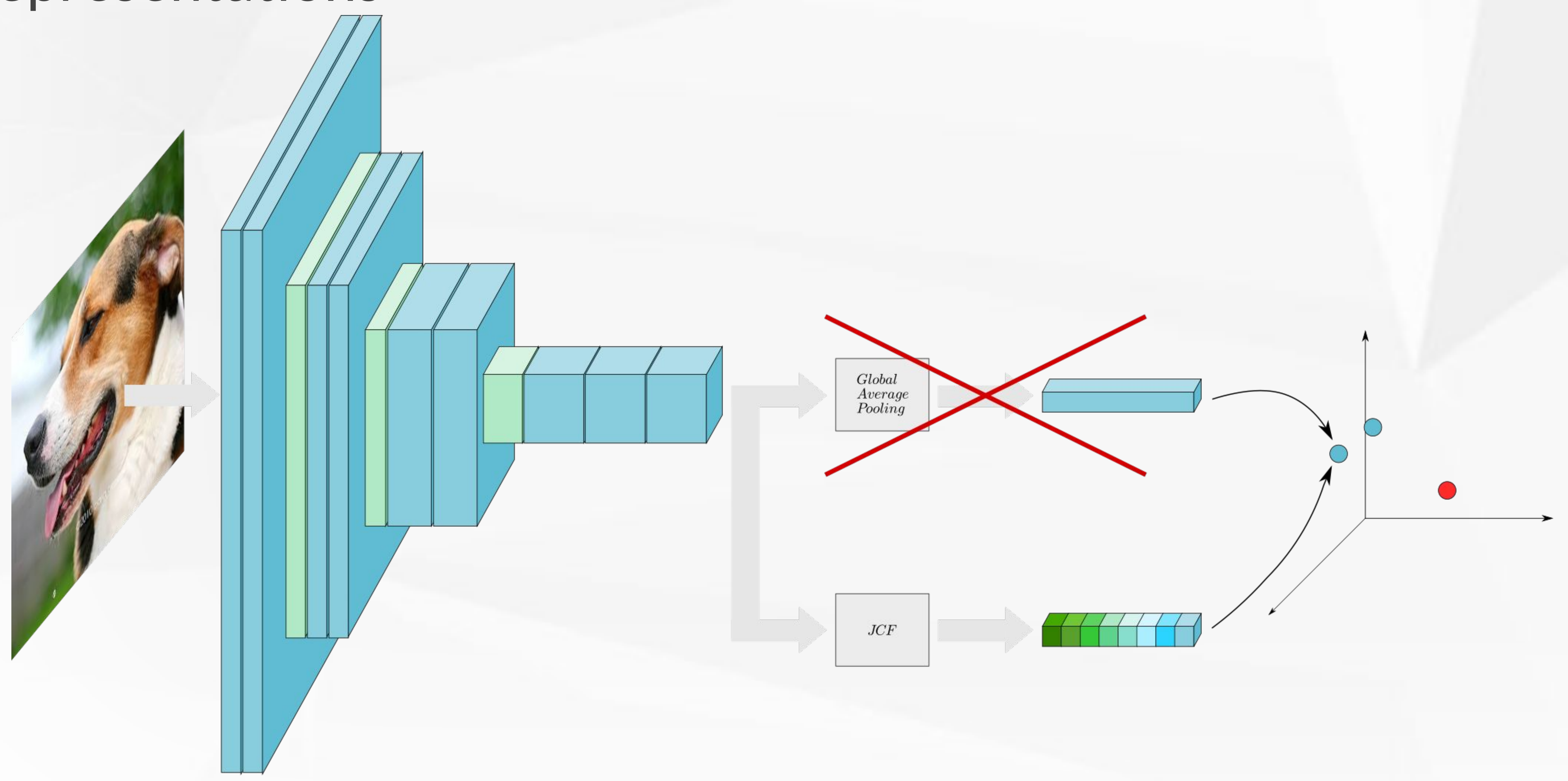
Problems

- **How to build rich and compact representations ?**
  - **Fine-grained** visual tasks => **Second-order** pooling
  - **Unrelated** feature aggregation => **Codebook** strategies
  - **Compactness** in representation learning => Deep **metric** learning (DML)
- **What about the drawbacks ?**
  - Second-order **dimensionality** is too large for DML
  - Codebook strategies further increase this **dimensionality**

Propositions

Joint Codebook-Factorization strategy

- Codebook + second-order pooling increases performances, albeit at a prohibitive cost
- Joint factorization and codebook strategy => rich and compact representations



Joint Codebook Factorization

Raw representation

- We duplicate the codebook assignment for symmetry purpose:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \otimes \mathbf{x} \otimes \mathbf{h}(\mathbf{x}) \otimes \mathbf{x}$$

Raw projection

- The i-th dimension is computed as follows:

$$z_i = \left\langle \mathbf{w}_i \; ; \; \mathbf{h}(\mathbf{x}) \otimes \mathbf{x} \otimes \mathbf{h}(\mathbf{x}) \otimes \mathbf{x} \right\rangle$$

First factorization

- Rank-one factorization to split each pair of codebook assignment and feature:

$$z_i = \left\langle \mathbf{p}_i \; ; \; \mathbf{h}(\mathbf{x}) \otimes \mathbf{x} \right\rangle \left\langle \mathbf{q}_i \; ; \; \mathbf{h}(\mathbf{x}) \otimes \mathbf{x} \right\rangle$$

Second factorization

- Multi-rank factorization which generalize intra-projection in codebook strategies:

$$\mathbf{p}_i = \sum_j \mathbf{e}^{(j)} \otimes \mathbf{u}_{i,j} \qquad \mathbf{q}_i = \sum_j \mathbf{e}^{(j)} \otimes \mathbf{v}_{i,j}$$

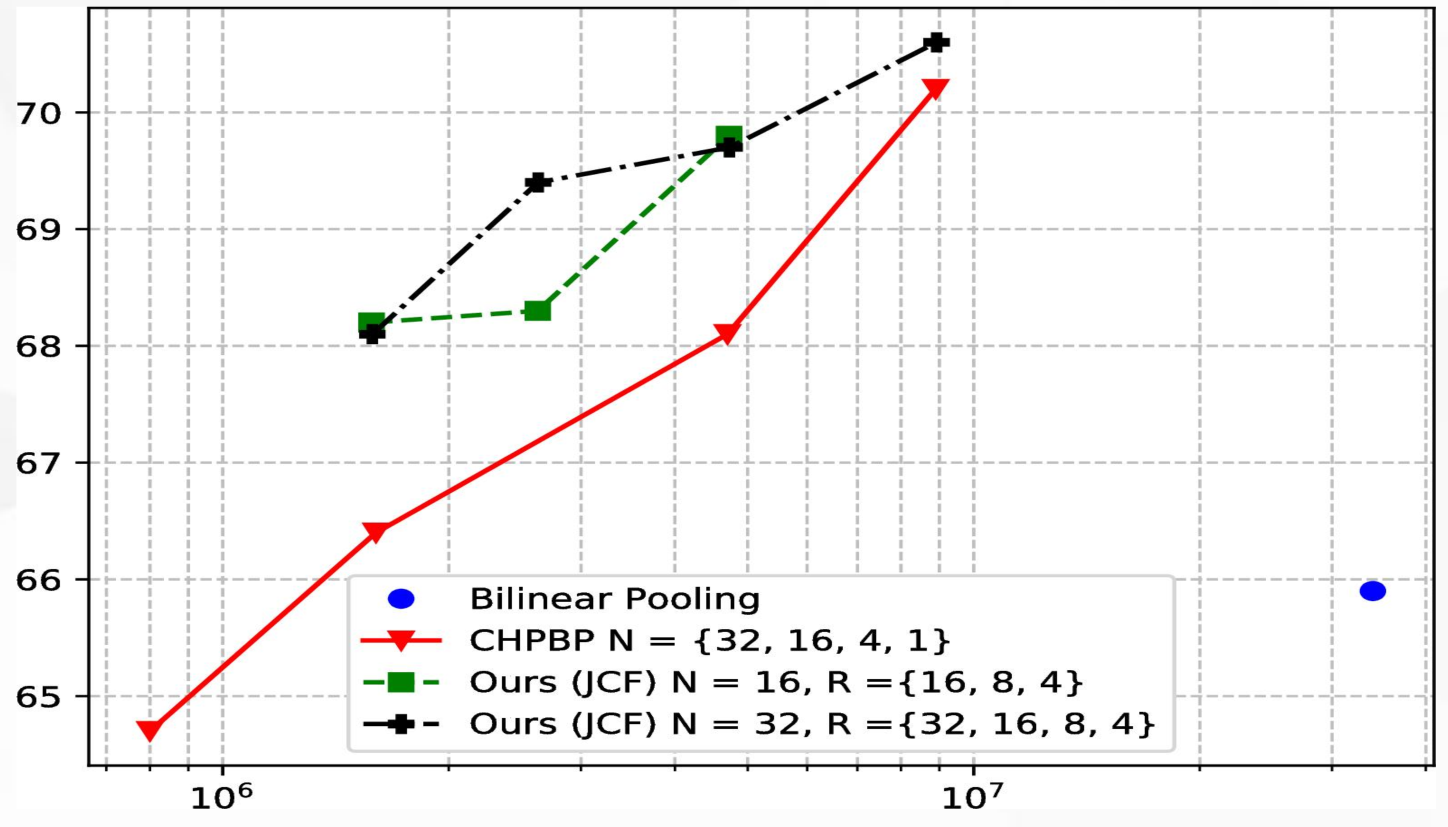
$$z_i = \left(\mathbf{h}(\mathbf{x})^\top \mathbf{U}_i^\top \; \mathbf{x}\right) \left(\mathbf{h}(\mathbf{x})^\top \mathbf{V}_i^\top \; \mathbf{x}\right)$$

Sharing projectors

- Entries in projection matrices can be shared between codebook entries:

$$z_i = \left(\mathbf{h}(\mathbf{x})^\top \mathbf{A} \widetilde{\mathbf{U}}_i^\top \; \mathbf{x}\right) \left(\mathbf{h}(\mathbf{x})^\top \mathbf{B} \widetilde{\mathbf{V}}_i^\top \; \mathbf{x}\right)$$

Results



Recall@1 against number of parameters

Method	CUB	CARS	SOP
HTL	57.1	81.4	74.8
JCF-32	**60.1**	**82.6**	**77.4**
JCF-32-8	58.1	74.2	76.6

Comparison to the state-of-the-art on three image retrieval datasets

Conclusion

- Codebook strategies further improve second-order pooling
- Joint codebook-factorization greatly reduces the representation size with fewer loss in performances
- Decrease in parameters can be handled by sharing projectors at the cost of a drop in performances

Acknowledgments

We would like to acknowledge COMUE Paris Seine University, University of Cergy-Pontoise and M2M Factory for their financial and technical support.

Contact us: {pierre.jacob, aymeric.histace}@ensea.fr, david.picard@enpc.fr

etis — Université / / Paris Seine — Université de Cergy-Pontoise — ENSEA — CNRS