Sparse Coding for Third-order Super-symmetric Tensor Descriptors with Application to Texture Recognition

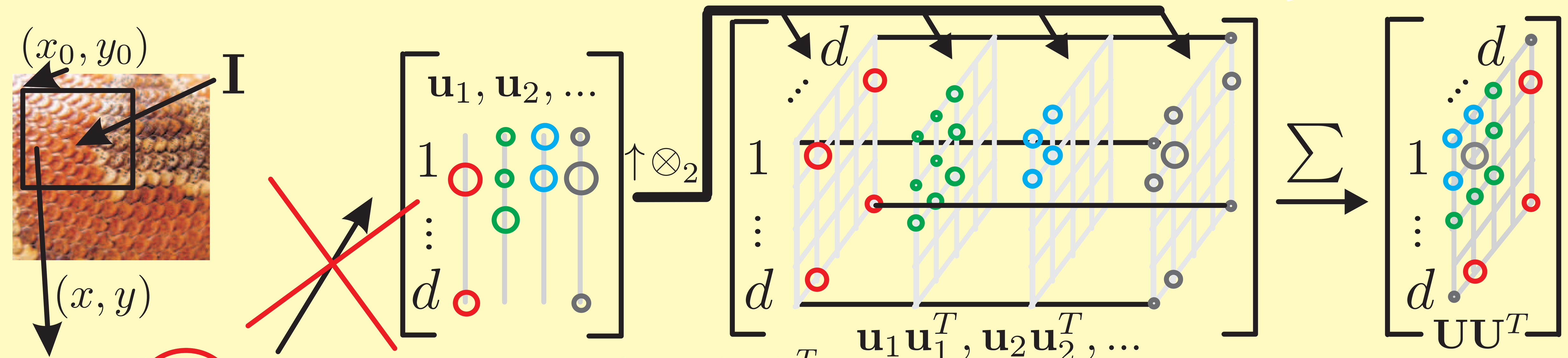
Piotr.Koniusz@nicta.com.au — National ICT Australia (NICTA), Data61/CSIRO

Anoop.Cherian@anu.edu.au — Australian National University, Canberra

Contributions and key ideas

- We propose novel Third-order Super-symmetric Tensor (TOSST) Texture Descriptors.
- We embed low-level features used for constructing TOSST into Reproducing Kernel Hilbert Space.
- Evaluating kernels is costly. We linearize our kernel formulation to obtain TOSST descriptors.
- TOSST representations $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d$ are cubic w.r.t. d – the size of features used to construct them.
- Therefore, we propose Sparse Coding for Third-order Super-symmetric Tensor descriptors to learn a tensor dictionary and code TOSST with $K \ll d^3$ atoms.
- However, the K dictionary atoms are themselves tensors $\boldsymbol{\mathcal{B}} \in \mathfrak{S}^d$ – overparametrized and costly to learn. Instead, we learn K matrix and vector atom pairs $\mathbf{B} \in \mathcal{S}_+^d$ and $\mathbf{b} \in \mathbb{R}_+^d$. Each tensor atom $\boldsymbol{\mathcal{B}}$ is low-rank approximated by the outer product of matrix $\mathbf{B}$ and vector $\mathbf{b}$, *i.e.* $\boldsymbol{\mathcal{B}}_k \approx \mathbf{B}_k \uparrow\otimes \mathbf{b}_k, \forall k$.

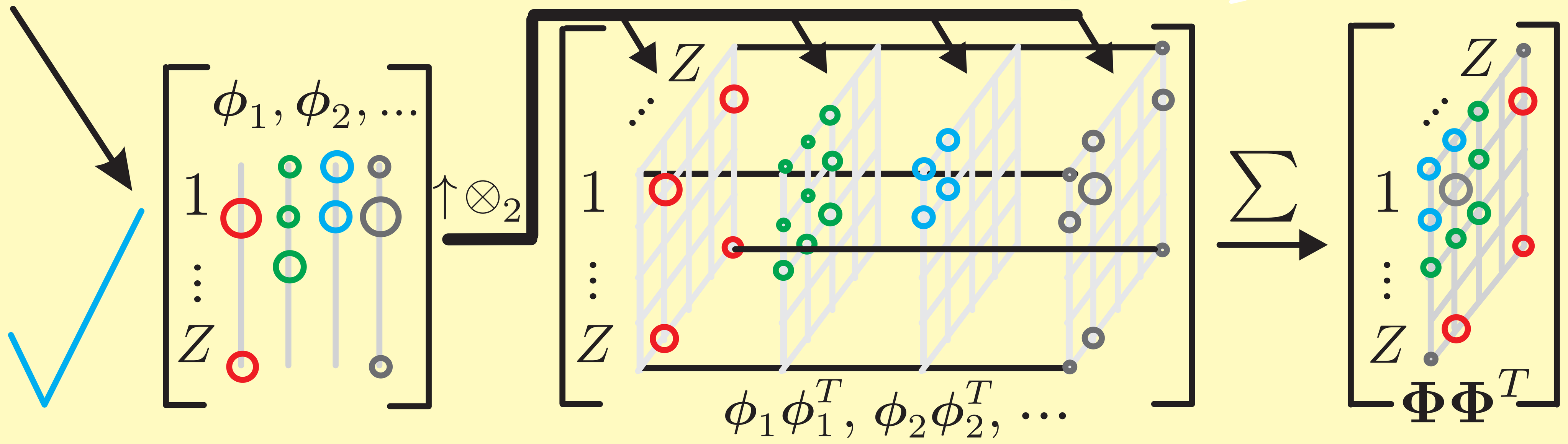
Third-order Super-symmetric Tensor (TOSST) descriptors



$$\mathbf{u}_{xy}(\mathbf{I}) = \left[\frac{x-x_0}{w-1}, \frac{y-y_0}{h-1}, I_{xy}, \frac{\partial I_{xy}}{\partial x}, \frac{\partial I_{xy}}{\partial y}, \frac{\partial^2 I_{xy}}{\partial x^2}, \frac{\partial^2 I_{xy}}{\partial y^2}\right]^T$$

$\phi: \mathbb{R} \rightarrow \mathbb{R}^Z$ (feature maps) where $G_\sigma(\mathbf{u}-\bar{\mathbf{u}}) \approx \langle \boldsymbol{\phi}(\mathbf{u}), \boldsymbol{\phi}(\bar{\mathbf{u}})\rangle$

$$\boldsymbol{\phi}_{xy}(\mathbf{I}) = \left[\phi\left(\frac{x-x_0}{w-1}\right), \phi\left(\frac{y-y_0}{h-1}\right), \phi(I_{xy}), \phi\left(\frac{\partial I_{xy}}{\partial x}\right), \phi\left(\frac{\partial I_{xy}}{\partial y}\right), \phi\left(\frac{\partial^2 I_{xy}}{\partial x^2}\right), \phi\left(\frac{\partial^2 I_{xy}}{\partial y^2}\right)\right]^T$$



- Gaussian between $\mathbf{u} \in \mathbb{R}^{d'}$ and $\bar{\mathbf{u}} \in \mathbb{R}^{d'}$ can be rewritten as:

$$G_\sigma(\mathbf{u}-\bar{\mathbf{u}}) = e^{-\|\mathbf{u}-\bar{\mathbf{u}}\|_2^2/2\sigma^2} = \left(\frac{2}{\pi\sigma^2}\right)^{\frac{d'}{2}} \int_{\boldsymbol{\zeta}\in\mathbb{R}^{d'}} G_{\sigma/\sqrt{2}}(\mathbf{u}-\boldsymbol{\zeta})\, G_{\sigma/\sqrt{2}}(\bar{\mathbf{u}}-\boldsymbol{\zeta})\, d\boldsymbol{\zeta}.$$

- Finite approximation by $\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_Z$ pivots is:

$$\boldsymbol{\phi}(\mathbf{u}) = \left[G_{\sigma/\sqrt{2}}(\mathbf{u}-\boldsymbol{\zeta}_1), \ldots, G_{\sigma/\sqrt{2}}(\mathbf{u}-\boldsymbol{\zeta}_Z)\right]^T,$$

$$\text{and } G_\sigma(\mathbf{u}-\bar{\mathbf{u}}) \approx \left\langle \frac{\boldsymbol{\phi}(\mathbf{u})}{\|\boldsymbol{\phi}(\mathbf{u})\|_2}, \frac{\boldsymbol{\phi}(\bar{\mathbf{u}})}{\|\boldsymbol{\phi}(\bar{\mathbf{u}})\|_2} \right\rangle.$$

- We linearize the sum kernel on τ low-level features:

$$K\left(\mathbf{I}^a_{xy}, \mathbf{I}^b_{x'y'}\right) = \sum_{i=1}^{\tau} G_{\sigma_i}\left(\phi^i(\mathbf{I}^a_{xy}) - \phi^i(\mathbf{I}^b_{x'y'})\right) \approx \left\langle \boldsymbol{\phi}^a_{xy}, \boldsymbol{\phi}^b_{x'y'} \right\rangle.$$

- We use $K^r(\mathbf{I}^a_{xy}, \mathbf{I}^b_{x'y'}) \approx \left\langle \boldsymbol{\phi}^a_{xy}, \boldsymbol{\phi}^b_{x'y'} \right\rangle^r$ – polynomial ker. of order $r=3$. We expand it over regions $\mathcal{R}$:

$$\sum_{\substack{(x,y)\in\mathcal{R}^a\\(x',y')\in\mathcal{R}^b}} K^r(\mathbf{I}^a_{xy}, \mathbf{I}^b_{x'y'}) \approx \left\langle \underset{(x,y)\in\mathcal{R}^a}{\text{Avg}} \left(\otimes_r \boldsymbol{\phi}^a_{xy}\right), \underset{(x'y')\in\mathcal{R}^b}{\text{Avg}} \left(\otimes_r \boldsymbol{\phi}^b_{x'y'}\right) \right\rangle.$$

- We hope to release some code in 6-8 weeks. Check: http://claret.wikidot.com.

Sparse Coding for Third-order Super-symmetric Tensor descriptors

- Tensors $\boldsymbol{\mathcal{X}}$ are cubic w.r.t. size of low-level features. To compress them, we learn a dictionary:

$$\underset{\substack{\boldsymbol{\mathcal{B}}_1,\ldots,\boldsymbol{\mathcal{B}}_K\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N}}{\arg\min} \sum_{n=1}^{N} \left\| \boldsymbol{\mathcal{X}}_n - \sum_{k=1}^{K} \boldsymbol{\mathcal{B}}_k \alpha_k^n \right\|_F^2 + \lambda \|\boldsymbol{\alpha}^n\|_1.$$

- Resulting sparse codes $\boldsymbol{\alpha}$ from regions are pooled and used for SVM training.
- However, $\boldsymbol{\mathcal{B}}$ is also overparametrized. Therefore, we learn a low-rank dictionary:

$$\underset{\substack{\mathbf{B}_1,\ldots,\mathbf{B}_K\\ \mathbf{b}_1,\ldots,\mathbf{b}_K\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N}}{\arg\min} \sum_{n=1}^{N} \left\| \boldsymbol{\mathcal{X}}_n - \sum_{k=1}^{K} (\mathbf{B}_k \uparrow\otimes \mathbf{b}_k)\, \alpha_k^n \right\|_F^2 + \lambda \|\boldsymbol{\alpha}^n\|_1.$$

- We introduce optimization variables β_k^n and rewrite the loss function by taking the slices out:

$$\underset{\substack{\boldsymbol{\mathcal{B}}\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N}}{\arg\min} \sum_{n=1}^{N} \sum_{s=1}^{S} \left\| \mathbf{X}_n^s - \sum_{k=1}^{K} \mathbf{B}_k \beta_k^{s,n} \right\|_F^2 + \lambda \|\boldsymbol{\alpha}^n\|_1,$$

subject to $\boldsymbol{\beta}_k^n = \mathbf{b}_k \alpha_k^n, \quad \forall k \in \mathcal{I}_K$ and $n \in \mathcal{I}_N$.

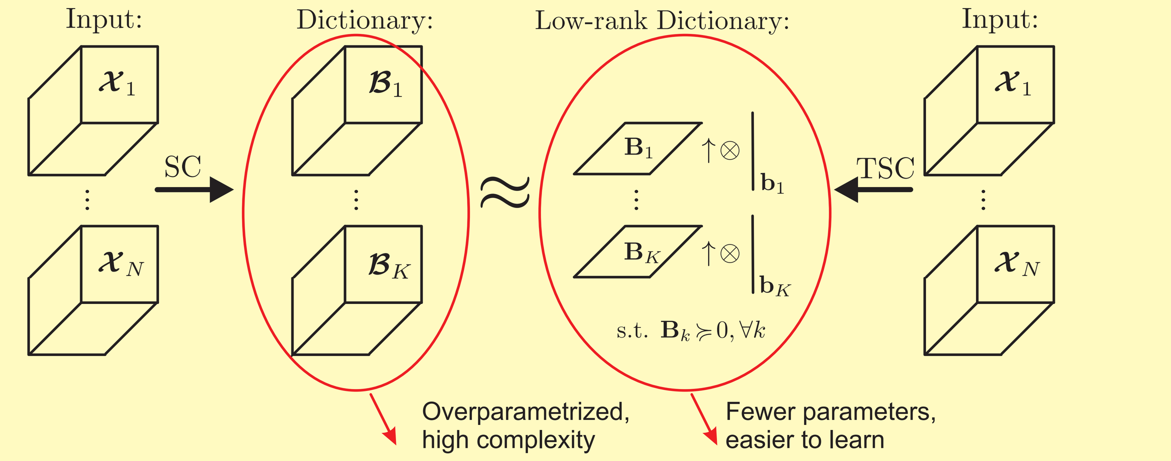
- We introduce chosen constraints and rewrite our tensor dictionary learning and sparse coding as:

$$\underset{\substack{\boldsymbol{\mathcal{B}}\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N \geq 0}}{\arg\min} \sum_{n=1}^{N} \sum_{s=1}^{S} \left\| \mathbf{X}_n^s - \sum_{k=1}^{K} \mathbf{B}_k \beta_k^{s,n} \right\|_F^2 + \gamma \left\| \boldsymbol{\beta}^{s,n} - \mathbf{b}^s \odot \boldsymbol{\alpha}^n \right\|_2^2 + \lambda \|\boldsymbol{\alpha}^n\|_1,$$

subject to $\mathbf{B}_k \succcurlyeq 0,\ \text{Tr}(\mathbf{B}_k) \leq 1,\ \|\mathbf{b}_k\|_1 \leq 1,\ k \in \mathcal{I}_K$.

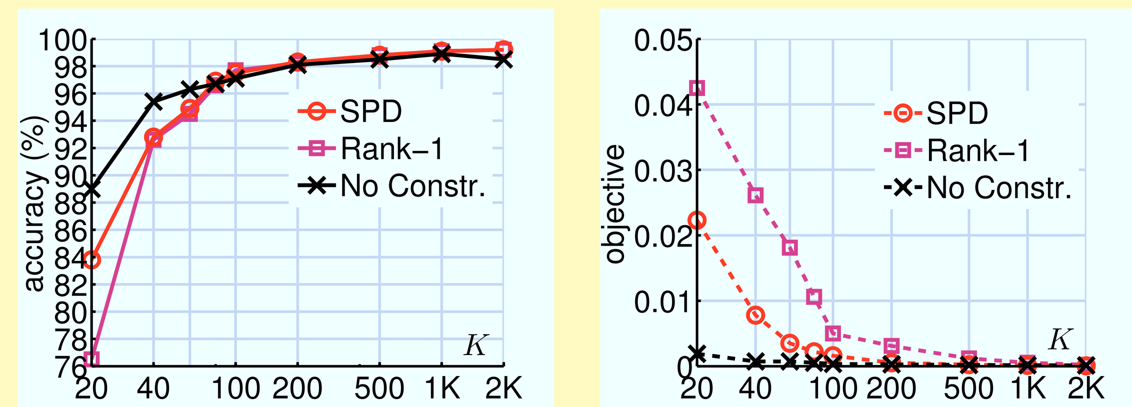
- Let $\widetilde{\boldsymbol{\mathcal{X}}}$ be the sparse approximation to $\boldsymbol{\mathcal{X}}$, then:

$$\text{TRank}(\widetilde{\boldsymbol{\mathcal{X}}}) \leq \min\left(\left| \bigcup_{s=1}^{d} \text{Supp}(\mathbf{X}^s) \right| \text{Rank}(\mathbf{B}),\ d^2 \right).$$

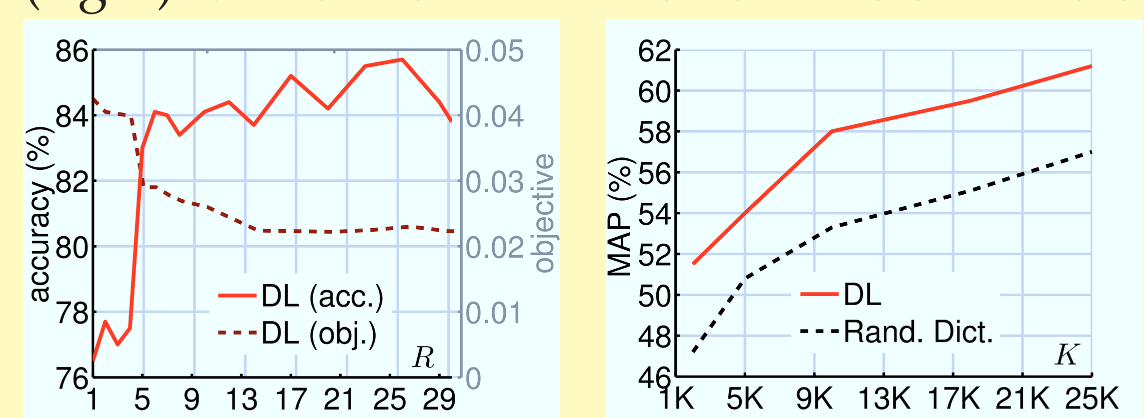


Results

- Brodatz textures: performance vs. dictionary size:



- (Left) performance vs. rank R, $K=20$, (right) learned vs. rand. dict. on PASCAL VOC07:



	(i) linear, no lum.	(ii) linear, lum.	(iii) RBF, no lum.	(iv) RBF, lum.	(v) RBF, opp.	Brodatz	UIUC materials
	Brodatz	Brodatz	Brodatz	Brodatz	UIUC materials	ELBCM 98.72% [4]	RSR 52.8±5.1% [5]
d	6	7	30	35	45	L^2ECM 97.9%	CDL 52.3±4.3%
acc.	93.9±0.2%	99.4±0.1%	99.4±0.2%	**99.9±0.08%**	**58.0±4.3%**	RC 97.7%	SD 43.5%

Eigenvalue Power Normalization

- $K(\mathcal{R}^a\!, \mathcal{R}^b) \approx \left\langle \underset{(x,y)\in\mathcal{R}^a}{\text{Avg}} \left(\otimes_r \boldsymbol{\phi}^a_{xy}\right), \underset{(x'y')\in\mathcal{R}^b}{\text{Avg}} \left(\otimes_r \boldsymbol{\phi}^b_{x'y'}\right) \right\rangle = \langle \boldsymbol{\mathcal{V}}^a\!, \boldsymbol{\mathcal{V}}^b \rangle.$
- Use better similarity for PSD matrices/tensors: $K^*(\mathcal{R}^a\!, \mathcal{R}^b) = \langle \boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{V}}^a), \boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{V}}^b) \rangle = \langle \boldsymbol{\mathcal{X}}^a\!, \boldsymbol{\mathcal{X}}^b \rangle$, *e.g.* ePN:

$$(\boldsymbol{\mathcal{E}}; \boldsymbol{A}_1, \ldots, \boldsymbol{A}_r) = \text{HOSVD}(\boldsymbol{\mathcal{V}})$$
$$\hat{\boldsymbol{\mathcal{E}}} = \text{Sgn}(\boldsymbol{\mathcal{E}})\, |\boldsymbol{\mathcal{E}}|^{\gamma}$$
$$\hat{\boldsymbol{\mathcal{V}}} = ((\hat{\boldsymbol{\mathcal{E}}} \otimes_1 \boldsymbol{A}_1) \ldots) \otimes_r \boldsymbol{A}_r$$
$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{V}}) = \text{Sgn}(\hat{\boldsymbol{\mathcal{V}}})\, |\hat{\boldsymbol{\mathcal{V}}}|^{\gamma^*}$$

References

[1] Higher-order Occurrence Pooling for Bags-of-Words: Visual Concept Detection. P. Koniusz, F. Yan, P. H. Gosselin, K. Mikolajczyk. TPAMI 2016.

[2] Tensor Representations via Kernel Linearlization for Action Recognition from 3D Skeletons. P. Koniusz, A. Cherian, F. Porikli. ArXiv 2016.

[3] Convolutional Kernel Networks. J. Mairal, P. Koniusz, Z. Harchaoui, C. Schmid. NIPS 2014.

[4] Enhanced local binary covariance matrices (ELBCM) for texture analysis and object tracking. A. Romero, M. Y. Terán, M. Gouiffès, L. Lacassagne. MIRAGE 2013.

[5] Sparse Coding and Dictionary Learning for Symmetric Positive Definite Matrices: A Kernel Approach. M. Harandi, C. Sanderson, R. Hartley, B. Lovell. ECCV 2012.