High-order Tensor Pooling with Attention for Action Recognition

Lei Wang[1,2] Ke Sun[2,1] Piotr Koniusz[2,1]

[1]Australian National University
[2]Data61/CSIRO

April 9, 2024

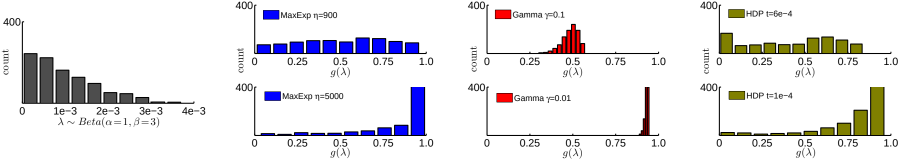
Motivation

Figure 1: CNN filters respond differently to tree leaf stimuli across spatial regions. **Detecting a leaf reliably predicts a tree's presence**.

Figure 2: Differing **feature counts** challenge classifier generalization. Training with few leaves may lead to misclassification of images with thousands, as boundaries are **sensitive** to observed features.

Motivation (cont.)

Higher-order representations undergo a non-linearity such as **Power Normalization (PN)**: reduce/boost contributions from frequent/infrequent visual stimuli in an image, respectively.



(a) Initial spectral dist. (b) MaxExp (c) Gamma (d) HDP

Figure 3: The intuitive principle of the **Eigenvalue Power Normalization (EPN)**.

- Given a discrete eigenspectrum following a Beta distribution, the pushforward distribution of MaxExp and HDP are very similar.
- For small γ, Gamma is also similar to MaxExp and HDP.
- Note that all three EPN functions **whiten the spectrum** (map the majority of values to be $\sim$1) thus removing burstiness (acting as a **spectral detector**).
- As EPN prevents burstiness, it replaces counting correlated features with detecting them, thus being invariant to their spatial/temporal extent.

HoT with EPN

EPN performs a spectrum transformation on $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times d_2 \ldots \times d_r}$:

$$(\boldsymbol{\lambda}; \boldsymbol{U}_1, ..., \boldsymbol{U}_r) = \text{HOSVD}(\boldsymbol{\mathcal{X}}), \tag{1}$$

$$\hat{\boldsymbol{\lambda}} = g(\boldsymbol{\lambda}), \tag{2}$$

$$\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{X}}) = ((\hat{\boldsymbol{\lambda}} \times_1 \boldsymbol{U}_1) \ldots) \times_r \boldsymbol{U}_r, \tag{3}$$

- Let $\boldsymbol{\Phi} \equiv \{\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N \in \mathbb{R}^d\}$ be feature vectors extracted from an instance to classify, *e.g.*, video sequences, images, text documents, *etc*.
- EPN retrieves factors which quantify whether there is **at least one** datapoint $\boldsymbol{\phi}_n$, $n \in \mathcal{I}_N$, projected into each subspace spanned by r-tuples of eigenvector from matrices $\boldsymbol{U}_1 = \boldsymbol{U}_2 = \ldots = \boldsymbol{U}_r$.
- For brevity, assume order $r=3$, a super-symmetric tensor, and **any** 3**-tuple of eigenvectors** $\mathbf{u}$, $\mathbf{v}$, and $\boldsymbol{w}$ from $\boldsymbol{U}$.
- Note that $\mathbf{u} \perp \mathbf{v}, \mathbf{v} \perp \boldsymbol{w}$ and $\mathbf{u} \perp \boldsymbol{w}$ due to orthogonality of eigenvectors for super-symmetric tensors, *e.g.*, $\mathbf{U}\boldsymbol{\lambda}^{\ddagger}\mathbf{V} = [\boldsymbol{\mathcal{X}}_{:,:,1}, ..., \boldsymbol{\mathcal{X}}_{:,:,d}] \in \mathbb{R}^{d \times d^2}$ where $\boldsymbol{\lambda}^{\ddagger}$ are eigenvalues of the **unfolded tensor** $\boldsymbol{\mathcal{X}}$.
- If we have d unique eigenvectors, we can enumerate $\binom{d}{r}$ r-tuples and thus $\binom{d}{r}$ subspaces $\mathbb{R}^{d \times r} \subset \mathbb{R}^{d \times d}$.

HoT with EPN (cont.)

Our **super-symmetric tensor descriptor** is:

$$\boldsymbol{\mathcal{X}} = \frac{1}{N} \sum_{n \in \mathcal{I}_N} \uparrow\otimes_r \boldsymbol{\phi}_n, \tag{4}$$

The 'diagonalization' of $\boldsymbol{\mathcal{X}}$ by eigenvectors $\mathbf{u}$, $\mathbf{v}$, and $\boldsymbol{w}$ produces core tensor:

$$\lambda_{\mathbf{u},\mathbf{v},\boldsymbol{w}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \boldsymbol{w}, \tag{5}$$

$\lambda_{\mathbf{u},\mathbf{v},\boldsymbol{w}}$ is **a coefficient** from the core tensor $\boldsymbol{\lambda}$. Combining Eq. (4) & (5) yields:

$$\begin{aligned} \lambda_{\mathbf{u},\mathbf{v},\boldsymbol{w}} &= \frac{1}{N} \sum_{n \in \mathcal{I}_N} \uparrow\otimes_3 \boldsymbol{\phi}_n \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \boldsymbol{w} \\ &= \frac{1}{N} \sum_{n \in \mathcal{I}_N} \langle \boldsymbol{\phi}_n, \mathbf{u} \rangle \langle \boldsymbol{\phi}_n, \mathbf{v} \rangle \langle \boldsymbol{\phi}_n, \boldsymbol{w} \rangle . \end{aligned} \tag{6}$$

- Let $\boldsymbol{\phi}_n$ be 'optimally' projected into subspace spanned by $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{w}$ when $\psi'_n = \langle \boldsymbol{\phi}_n, \mathbf{u} \rangle \langle \boldsymbol{\phi}_n, \mathbf{v} \rangle \langle \boldsymbol{\phi}_n, \boldsymbol{w} \rangle$ is **maximized**.
- As our $\mathbf{u}$, $\mathbf{v}$, and $\boldsymbol{w}$ are orthogonal w.r.t. each other and $||\boldsymbol{\phi}_n||_2 = 1$, simple Lagrange equation $\mathcal{L} = \Pi_{i=1}^r \boldsymbol{e}_i^T \boldsymbol{\phi}_n + \lambda(||\boldsymbol{\phi}_n||_2^2 - 1)$ yields **maximum of** $\kappa = (1/\sqrt{r})^r$ **at** $\boldsymbol{\phi}_n = [(1/\sqrt{r}), ..., (1/\sqrt{r})]^T$.
- For **each** $n \in \mathcal{I}_N$, we store $\psi_n = \psi'_n / \kappa$ in a so-called event vector $\boldsymbol{\psi}$.

HoT with EPN (cont.)

Assume $\boldsymbol{\psi}\in\{0,1\}^N$ stores N outcomes of drawing from Bernoulli distribution under the i.i.d. assumption: the probability p of an event $(\psi_n=1)$ & $1-p$ for $(\psi_n=0)$ are estimated by an expected value, $p=\operatorname{avg}_n \psi_n=\lambda_{\mathbf{u},\mathbf{v},\boldsymbol{w}}/\kappa$ $(0\leq\boldsymbol{\psi}\leq 1)$. The probability of at least one positive event $(\psi_n=1)$ projecting into the subspace spanned by r-tuples in N trials is:

$$\hat{\lambda}_{\mathbf{u},\mathbf{v},\boldsymbol{w}}=1-(1-p)^N=1-\left(1-\frac{\lambda_{\mathbf{u},\mathbf{v},\boldsymbol{w}}}{\kappa}\right)^N. \tag{7}$$

Each of $\binom{d}{r}$ subspaces spanned by r-tuples acts as a detector of projections into this subspace. Eq. (7) is the spectral MaxExp pooling with κ normalization. Considering the dot-product between EPN-norm. tensors $\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{X}})$ and $\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{Y}})$:

$$\begin{aligned}&\langle\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{X}}),\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{Y}})\rangle\\ &=\sum_{\substack{\mathbf{u}\in\mathbf{U}(\boldsymbol{\mathcal{X}})\\ \mathbf{v}\in\mathbf{V}(\boldsymbol{\mathcal{X}})\\ \boldsymbol{w}\in\boldsymbol{W}(\boldsymbol{\mathcal{X}})}}\;\sum_{\substack{\mathbf{u}'\in\mathbf{U}(\boldsymbol{\mathcal{Y}})\\ \mathbf{v}'\in\mathbf{V}(\boldsymbol{\mathcal{Y}})\\ \boldsymbol{w}'\in\boldsymbol{W}(\boldsymbol{\mathcal{Y}})}}\hat{\lambda}_{\mathbf{u},\mathbf{v},\boldsymbol{w}}\hat{\lambda}'_{\mathbf{u}',\mathbf{v}',\boldsymbol{w}'}\,\langle\mathbf{u},\mathbf{u}'\rangle\,\langle\mathbf{v},\mathbf{v}'\rangle\,\langle\boldsymbol{w},\boldsymbol{w}'\rangle\,.\end{aligned} \tag{8}$$

Hence, all subspaces of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ spanned by r-tuples (*e.g.*, $r=3$ as above) are compared against each other for alignment by the cosine distance.

Backpropagating through HOSVD and/or SVD

Let $\boldsymbol{M}^{\#} = \boldsymbol{M}\boldsymbol{M}^T = \mathbf{U}\boldsymbol{\lambda}\mathbf{U}^T$ be an SPD matrix with simple eigenvalues, *i.e.*, $\lambda_{ii} \neq \lambda_{jj}, \forall i \neq j$. Then $\mathbf{U}$ coincides also with the eigenvector matrix of tensor $\boldsymbol{\mathcal{X}}$ for the given unfolding. To compute the derivative of $\mathbf{U}$ (we drop the index) w.r.t. $\boldsymbol{M}$ (and thus $\boldsymbol{\mathcal{X}}$), one has to follow the chain rule:

$$\frac{\partial \mathbf{U}}{\partial M_{kl}} = \sum_{i,j} \frac{\partial \mathbf{U}}{\partial (\boldsymbol{M}\boldsymbol{M}^T)_{ij}} \cdot \frac{\partial (\boldsymbol{M}\boldsymbol{M}^T)_{ij}}{\partial M_{kl}},$$

$$\text{where} \quad \frac{\partial u_{ij}}{\partial \boldsymbol{M}^{\#}} = u_{ij}(\lambda_{jj}\mathbb{I} - \boldsymbol{M}^{\#})^{\dagger}. \tag{9}$$

For SVD, we simply have to backpropagate through the chain rule:

$$\frac{\partial \mathbf{U}\boldsymbol{\lambda}\mathbf{U}^T}{\partial X_{m'n'}} = 2\,\mathrm{Sym}\left(\frac{\partial \mathbf{U}}{\partial X_{m'n'}}\boldsymbol{\lambda}\mathbf{U}^T\right) + \mathbf{U}\frac{\partial \boldsymbol{\lambda}}{\partial X_{m'n'}}\mathbf{U}^T,$$

$$\text{where} \quad \mathrm{Sym}(\mathbf{X}) = \frac{1}{2}(\mathbf{X} + \mathbf{X}^T). \tag{10}$$

Let $\mathbf{X} = \mathbf{U}\boldsymbol{\lambda}\mathbf{U}^T$ be an SPD matrix with simple eigenvalues, *i.e.*, $\lambda_{ii} \neq \lambda_{jj}, \forall i \neq j$, and $\mathbf{U}$ contain eigenvectors of matrix $\mathbf{X}$, then one can apply $\frac{\partial \lambda_{ii}}{\partial X} = \mathbf{u}_i\mathbf{u}_i^T$ and $\frac{\partial u_{ij}}{\partial X} = u_{ij}(\lambda_{jj}\mathbb{I} - X)^{\dagger}$.

Application to Action Recognition

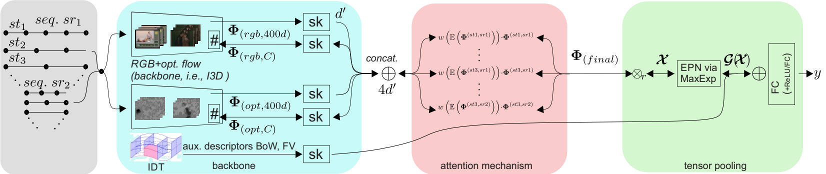


Figure 4: Our action recognition pipeline with the attention mechanism.

Our pipeline:

- extract subsequences (invariance to action localization)
- apply various sampling rates (invariance to action speed)
- extract 400D features (I3D pretrained on Kinetics-400)
- obtain intermediate matrices with feature vectors
- use count sketching (*sk*) to reduce dimensionality & concatenate features

Attention mechanism:

- The attention network $w:\mathbb{R}^{d'}\rightarrow\mathbb{R}$ outputs an attention score
- $\boldsymbol{\Phi}_w^{(i,j)}=w\left(\mathbb{E}\left(\boldsymbol{\Phi}^{(i,j)}\right)\right)\cdot\boldsymbol{\Phi}^{(i,j)}$, $i\in\{st_1,st_2,...\}$ & $j\in\{sr_1,sr_2,...,\}$
- form final feature matrix $\boldsymbol{\Phi}_{(final)}\in\mathbb{R}^{d\times N}$, $d=4d'$, then passed via Eq. (4).
- pass $\boldsymbol{\mathcal{X}}$ via EPN to obtain $\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{X}})\in\mathbb{R}^{d\times d\times d}$, one per instance to classify

Results & Discussions

SO+	*sp1*	*sp2*	*sp3*	mean	TO+	*sp1*	*sp2*	*sp3*	mean
(no EPN)	76.2	75.3	76.7	76.1	(no EPN)	75.4	74.0	75.0	74.8
HDP	81.4	78.8	80.1	80.1	HDP	81.8	79.6	81.3	80.9
MaxExp	81.7	79.1	80.1	80.3	MaxExp	82.3	79.9	81.2	81.1
MaxExp+IDT	86.1	85.2	85.8	**85.7**	MaxExp+IDT	87.4	86.7	87.5	**87.2**

ADL+I3D 81.5	Full-FT I3D 81.3	SCK(SO+) +IDT 85.1	SCK(TO+) +IDT 86.1

Table 1: (*top*) Our model *vs.* (*bottom*) SOTA on HMDB-51.

	static	*dynamic*	*mixed*	mean stat/dyn	mean all
SO+MaxExp	92.52	82.03	89.44	87.3	88.0
SO+MaxExp+IDT	94.92	86.63	96.02	90.8	92.5
TO+MaxExp+IDT	**95.36**	86.90	**97.04**	91.1	**93.1**
T-ResNet	92.41	81.50	89.00	87.0	87.6
ADL I3D	95.10	**88.30**	-	91.7	-

Table 2: (*top*) Our pipeline *vs.* (*bottom*) SOTA on YUP++.

	sp1	*sp2*	*sp3*	*sp4*	*sp5*	*sp6*	*sp7*	mAP
SO+MaxExp+IDT	75.7	82.5	79.4	75.1	75.7	76.8	75.9	77.3
TO+MaxExp+IDT	78.6	83.4	81.5	78.8	81.7	79.2	79.6	**80.4**

KRP-FS 70.0	KRP-FS+IDT 76.1	GRP 68.4	GRP+IDT 75.5

Table 3: (*top*) Our pipeline *vs.* (*bottom*) SOTA on MPII.

Thank you!