

Rank pooling and variants for action and activity recognition

Basura Fernando

ARC Centre of Excellence for Robotic Vision
Research School of Engineering
The Australian National University

Outline

- 1 Introduction
- 2 Dynamic Images
- 3 Hierarchical rank pooling
- 4 Learning discriminative dynamic

Collaborators



(a) Stephen Gould (ANU) (b) Tinne Tuytelaars (KUL) (c) Efstratios Gavves (UVA) (d) Andrea Vedaldi (OX) (e) Hakan Bilen ((OX))



(f) Marcus Hutter (ANU) (g) Jose Oramas (KUL) (h) Amir Godrati (KUL) (i) Peter Anderson (ANU)

Action recognition from video sequences



- A video is a sequence of n frames $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$.
- The frame at time t is represented by a vector $\mathbf{x}_t \in D$.
- The training set consists of N_{trn} number of videos from C number of action classes.
- Objective: Classify each test video to correct action class

Temporal pooling and encoding methods

- max-pooling - works well with CNN features
- sum-pooling - works well with Fisher vectors
- LSTM
- Temporal pyramids
- HMM, CRF, subspace-based methods

Temporal encoding with Rank Pooling

- Let V be a sequence of smoothed data
 $V = [\mathbf{v}_1 \succ \mathbf{v}_2 \succ \mathbf{v}_3 \cdots \succ \mathbf{v}_n]$ obtained from X
- Let \mathcal{D} be the dynamics of a video sequence V

Proposition : The dynamics of V can be approximated by linear function $\Phi_u = \Phi(V; \mathbf{u})$ parametrized by \mathbf{u}

$$\arg \min_u \|\mathcal{D} - \Phi_u\|. \quad (1)$$

- For a given definition of dynamics \mathcal{D} , there exists a family of functions Φ .
- What is a good family of such functions?

Learning to rank

- A pairwise linear ranking machine $\Phi(\mathbf{v}; \mathbf{u}) = \mathbf{u}^T \cdot \mathbf{v}$ learns parameter (\mathbf{u}) from the data such that
- $\forall t_i, t_j, \mathbf{v}_{t_i} \succ \mathbf{v}_{t_j} \iff \mathbf{u}^T \cdot \mathbf{v}_{t_i} > \mathbf{u}^T \cdot \mathbf{v}_{t_j}$

Using the structural risk minimization and max-margin framework, the objective is then to optimize

$$\begin{aligned} \operatorname{argmin}_{\mathbf{u}} \quad & \frac{1}{2} \|\mathbf{u}\|^2 + C \sum_{\forall i, j, \mathbf{v}_{t_i} \succ \mathbf{v}_{t_j}} \epsilon_{ij} \\ \text{s.t.} \quad & \mathbf{u}^T \cdot (\mathbf{v}_{t_i} - \mathbf{v}_{t_j}) \geq 1 - \epsilon_{ij} \\ & \epsilon_{ij} \geq 0. \end{aligned} \tag{2}$$

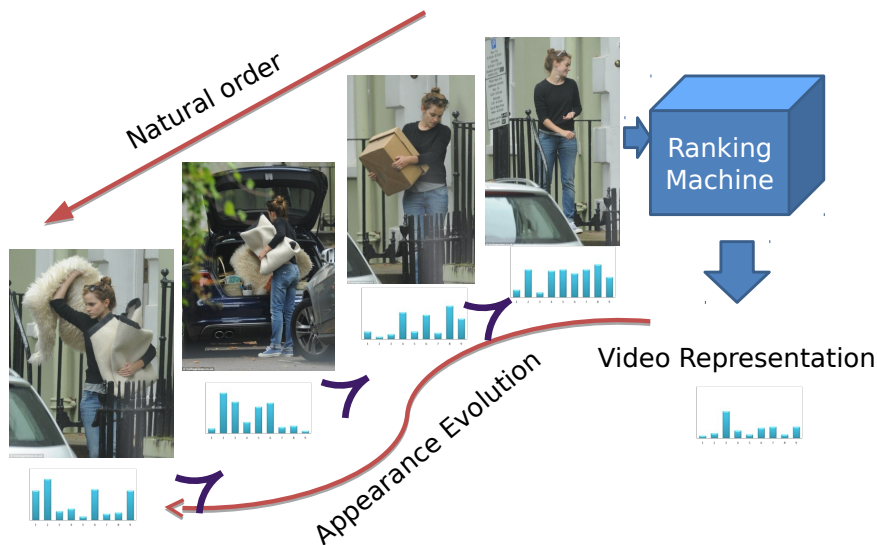
Observations:

- Parameter (\mathbf{u}) lies in the same space as the original data of V
- Parameter (\mathbf{u}) captures the information about ordering in V
- Parameter (\mathbf{u}) captures some information about the contents of V

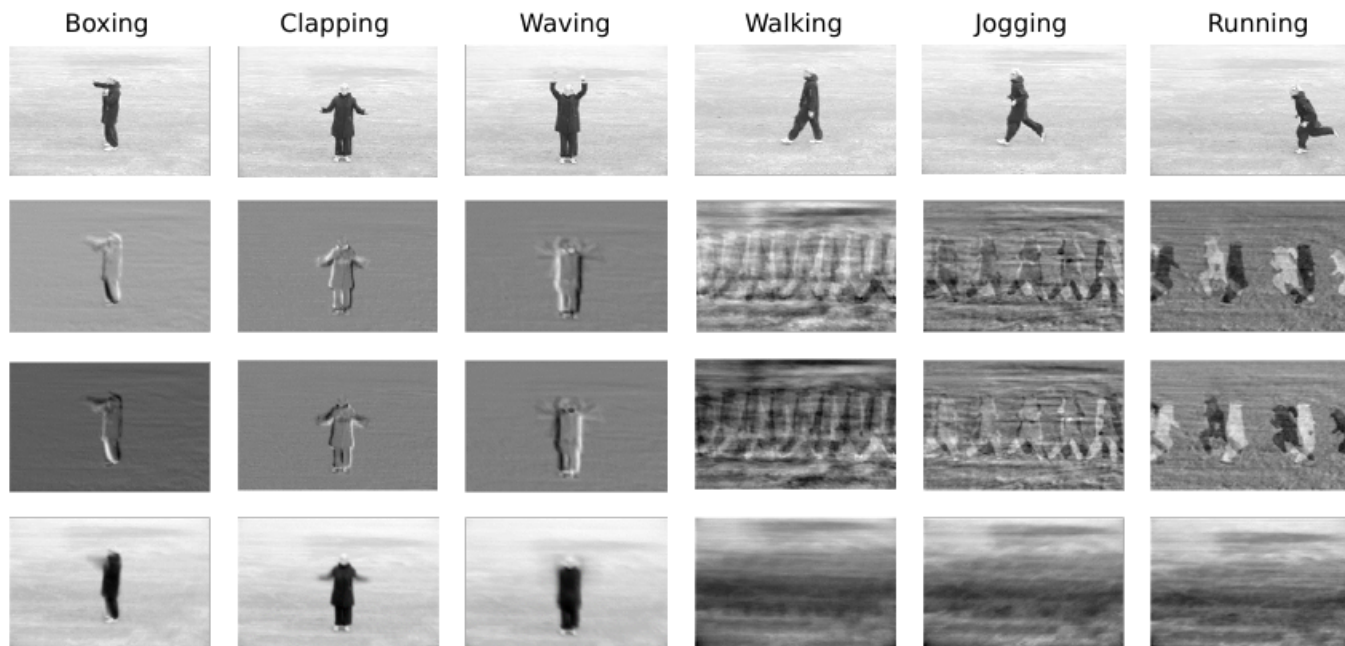
Rank pooling: Main idea

Proposition

We propose to use the parameters $\mathbf{u}_i \in D$ of Φ_i as a new video representation for capturing the specific *appearance evolution of the video* i.e. \mathcal{D}_i

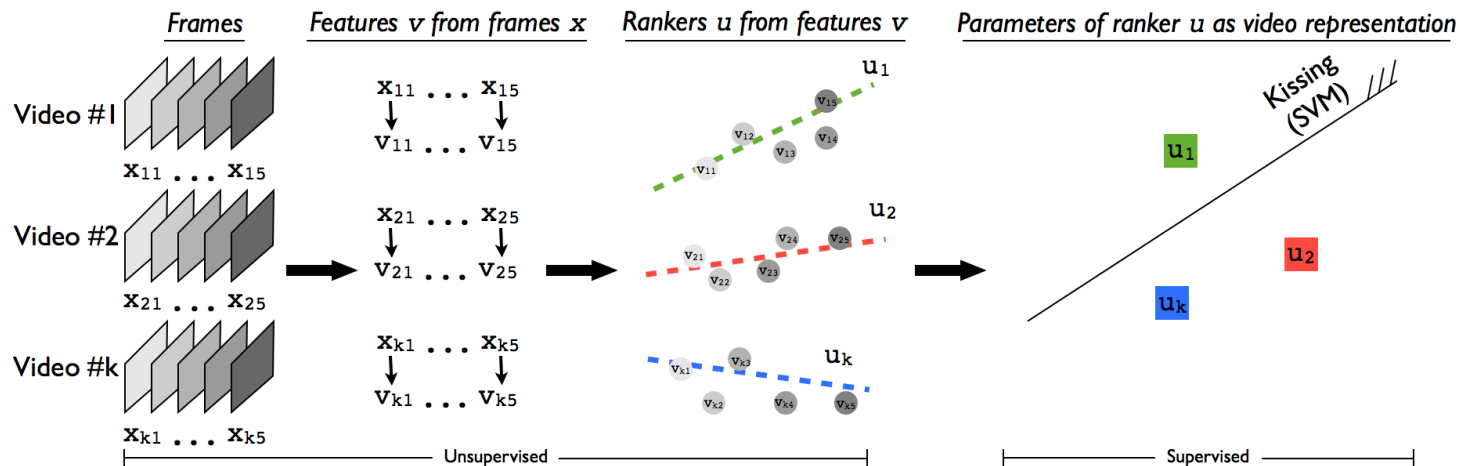


What is captured by rank pooling?



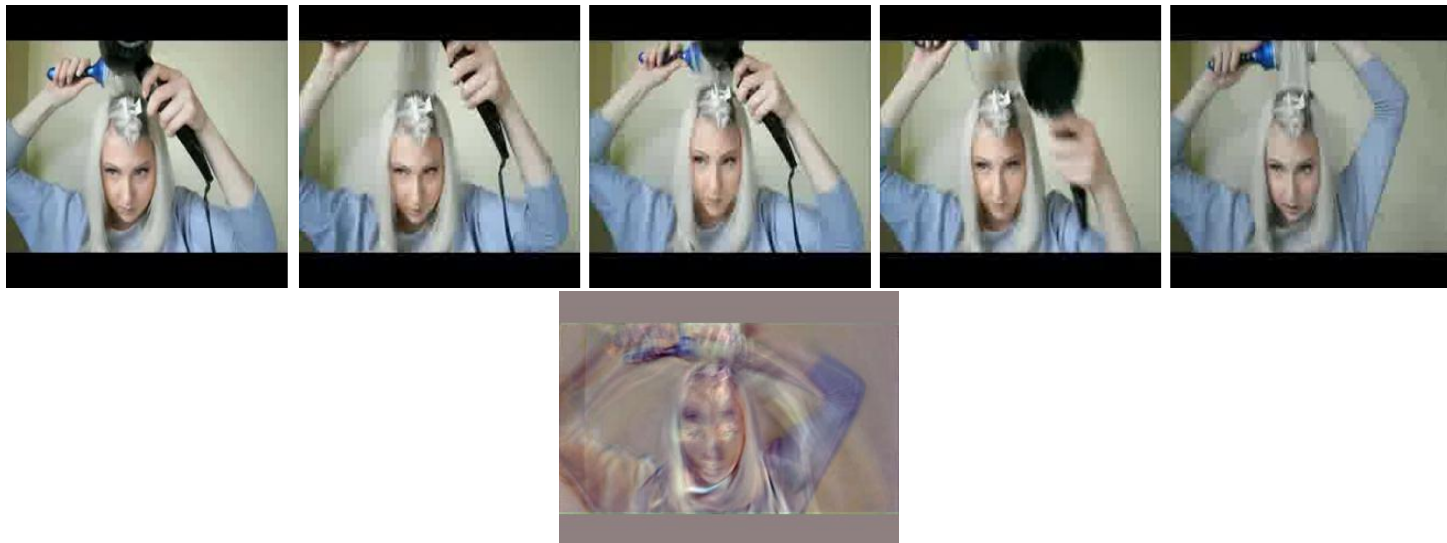
Rank Pooling - Algorithm

- 1 Extract dense or improved trajectory from the video
- 2 Fisher encode each frame using HOG, HOF, MBH features (create \mathbf{v}_{t_i})
- 3 Smooth the video signal X to obtain $V = [\mathbf{v}_1 \succ \mathbf{v}_2 \succ \mathbf{v}_3 \cdots \succ \mathbf{v}_n]$
- 4 Learns ranking function $\Phi(\mathbf{v}; \mathbf{u})$ from X
- 5 Represent each video with vector \mathbf{u}
- 6 Use standard classification framework for action classification.



- Forward and reverse Rank Pooling
- Non-linear Rank Pooling with non-linear feature maps
- Data augmentation with mirrored videos

What if we can summarize the motion information of a video into an single image?



Dynamic images of UCF 101

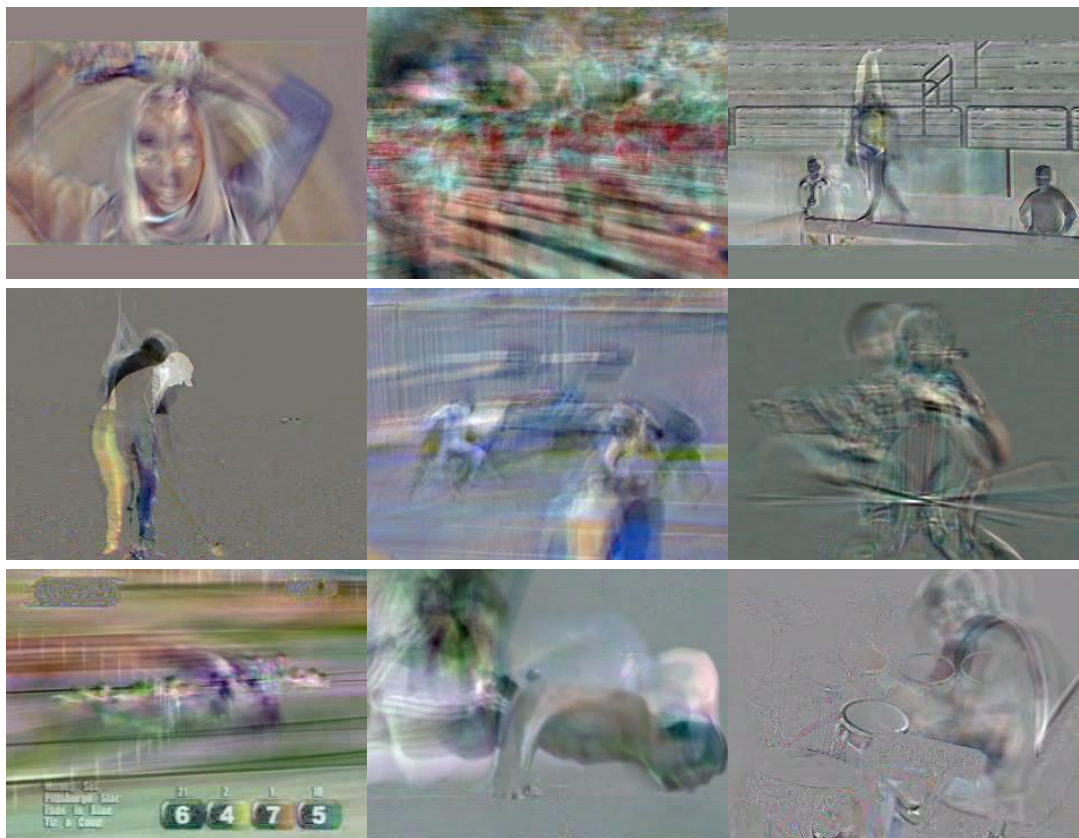
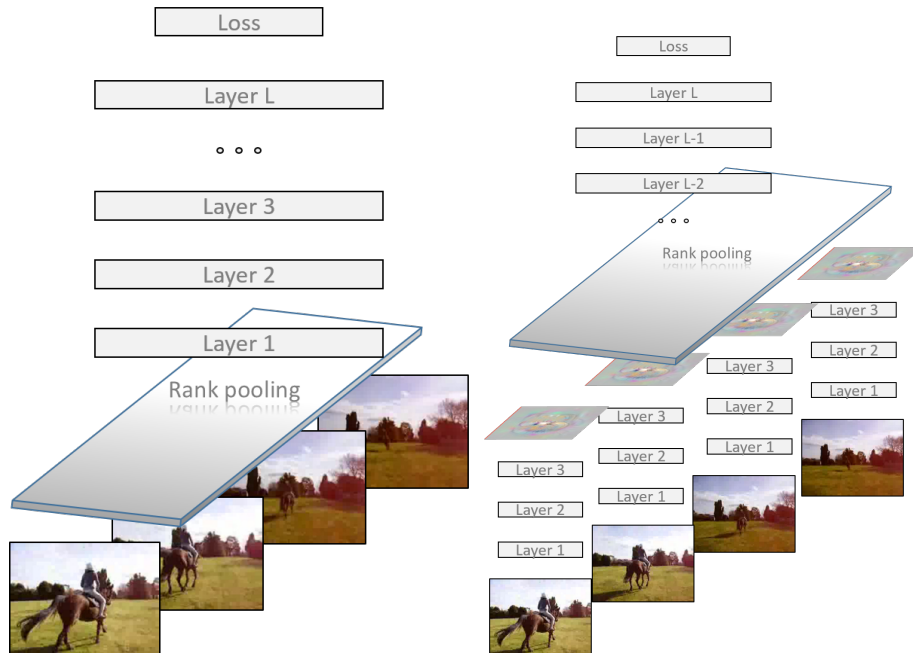


Figure 1 : Dynamic images summarizing the actions and motions that happen in images in standard 2d image format.

Learning dynamic images and dynamic maps

- End to end CNN training with approximate rank pooling
- Gradients of rank pooling is complex
- Need faster forward pass approximation



Dynamic image networks - temporal objective

- Let frames of a video be I_1, \dots, I_T .
- Let $\psi(I_t) \in \mathbb{R}^d$ be a representation for individual frame I_t in the video.
- Let $V_t = \frac{1}{t} \sum_{\tau=1}^t \psi(I_\tau)$ be time average of these features up to time t .
- The ranking function associates to each time t a score $S(t|\mathbf{u}) = \langle \mathbf{u}, V_t \rangle$, where $\mathbf{u} \in \mathbb{R}^d$ is a vector of parameters.

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\operatorname{argmin}} \rho(I_1, \dots, I_T; \psi) = \underset{\mathbf{u}}{\operatorname{argmin}} E(\mathbf{u}),$$
$$E(\mathbf{u}) = \frac{\lambda}{2} \|\mathbf{u}\|^2 + \frac{2}{T(T-1)} \times \sum_{q>t} \max\{0, 1 - S(q|\mathbf{u}) + S(t|\mathbf{u})\}. \quad (3)$$

Dynamic image networks - fast approximate rank pooling

- considers the first step in a gradient-based optimization of objective 6.
- initialize video specific parameters with $\mathbf{u} = \vec{0}$
- for arbitrary learning rate $\eta > 0$, new \mathbf{u} will be $\dot{\mathbf{u}} = \vec{0} - \eta \nabla E(\mathbf{u})|_{\mathbf{u}=\vec{0}} \propto -\nabla E(\mathbf{u})|_{\mathbf{u}=\vec{0}}$

$$\begin{aligned}\nabla E(\vec{0}) &\propto \sum_{q>t} \nabla \max\{0, 1 - S(q|\mathbf{u}) + S(t|\mathbf{u})\}|_{\mathbf{u}=\vec{0}} \\ &= \sum_{q>t} \nabla \langle \mathbf{u}, V_t - V_q \rangle = \sum_{q>t} V_t - V_q.\end{aligned}$$

We can further expand $\dot{\mathbf{u}}$ as follows

$$\dot{\mathbf{u}} \propto \sum_{q>t} V_q - V_t = \sum_{q>t} \left[\frac{1}{q} \sum_{i=1}^q \psi_i - \frac{1}{t} \sum_{j=1}^t \psi_j \right] = \sum_{t=1}^T \alpha_t \psi_t$$

where the coefficients α_t are given by

$$\alpha_t = 2(T - t + 1) - (T + 1)(H_T - H_{t-1}) \quad (4)$$

and where $H_t = \sum_{i=1}^t 1/i$ is the t -th Harmonic number (and where we define $H_0 = 0$). Hence the rank pooling operator reduces to

$$\hat{\rho}(I_1, \dots, I_T; \psi) = \sum_{t=1}^T \alpha_t \psi(I_t). \quad (5)$$

UCF101 dataset



Results on UCF101 & HMDB51 dataset

Method	SPLIT1	SPLIT2	SPLIT3	MEAN
Mean Image	52.6	53.4	51.7	52.6
Max Image	48.0	46.0	42.3	45.4
SDI	57.2	58.7	57.7	57.9

Table 1 : Comparing several video representative image models using UCF101

Method	HMDB51	UCF101
MDI	32.3	68.6
MDM(conv1)	–	67.1
MDI end-to-end	35.8	70.9

Table 2 : Evaluating the effect of end-to-end training for multiple dynamic images and multiple dynamic maps after the convolutional layer 1.

PART III. Hierarchical rank pooling

Need for hierarchical rank pooling

- Capacity of linear flat rank pooling is limited
- Dynamics of multiple granularities can be captured (low-level, mid-level, and high-level dynamics)
- Hierarchical networks of non-linear dynamic functions can be employed on sequence data

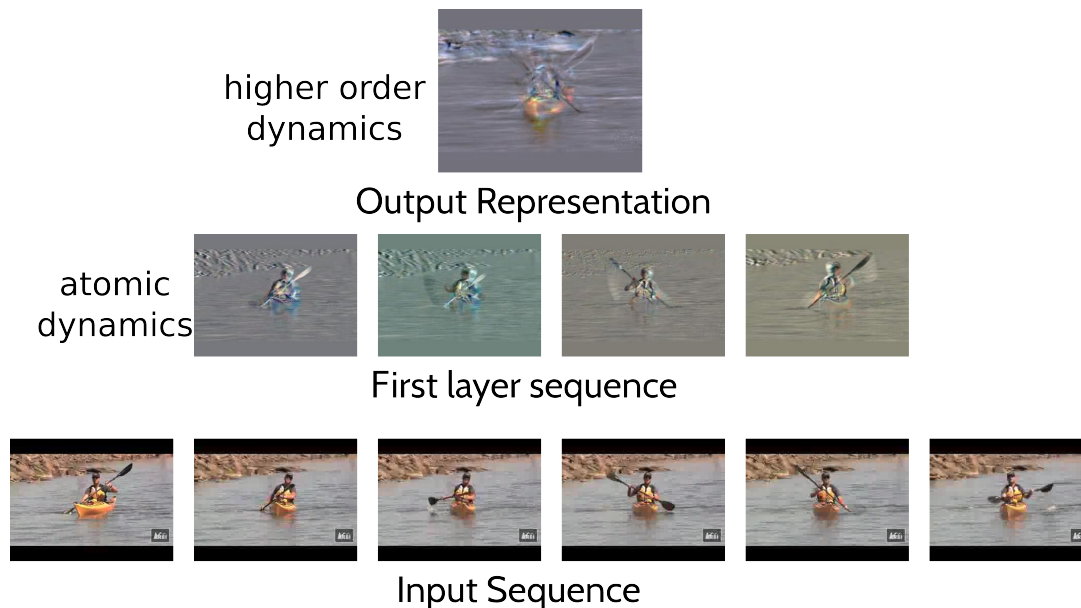


Figure 2 : Illustration of hierarchical rank pooling for encoding the

Hierarchical rank pooling - CVPR 2016

- Let $\Phi(x_n^m, x_{n+1}^m, \dots, x_{n+k}^m) \rightarrow x_k^{m+1}$ is the rank pooling function applied at layer m on k -th sequence
- $\Psi(x_n^m)$ is a non-linear feature encoding

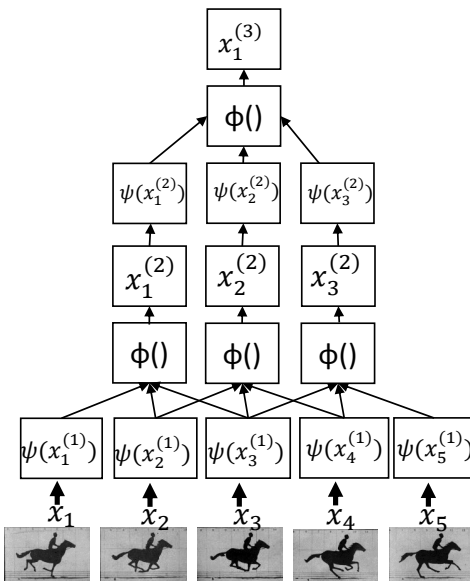


Figure 3 : Illustration of hierarchical temporal encoding networks

Hierarchical rank pooling $\Phi()$ and Ψ

Sequence encoding machinery ϕ

$$\Phi : \mathbf{u}^* \in \operatorname{argmin}_{\mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{u}\|^2 + \frac{C}{2} \sum_{t=1}^J \left[|t - \mathbf{u}^T \mathbf{v}_t| - \epsilon \right]_{\geq 0}^2 \right\} \quad (6)$$

Capturing non-linear dynamics ψ

$$\psi(x) = \left(\sqrt{\max\{0, x\}}, \sqrt{\max\{0, -x\}} \right) \quad (7)$$

Results - on baseline methods

METHOD	Hollywood2	HMDB51	UCF101
Average pooling	40.9	37.1	69.3
Max pooling	42.4	39.1	72.5
Tempo. pyramid (avg. pool)	46.5	39.1	73.3
Tempo. pyramid (max pool)	48.7	39.8	74.8
LSTM Srivastava2015	–	42.8	74.5
LRCN Donahue2015	–	–	68.8
Rank pooling	44.2	40.9	72.2
Recursive rank pooling	52.5	45.8	75.6
Hierarchical rank pooling	56.8	47.5	78.8
Improvement	+8.1	+4.7	+4.0

Table 3 : Comparing several temporal pooling methods for activity recognition using vgg-16’s fc6 features.

Comparing to state-of-the-art

	Hollywood2	HMDB51	UCF101
<i>our * method</i>	76.7	66.9	91.4
[Zha et al., 2015]	–	–	89.6
[Fernando et al., 2015]	73.7	63.7	–
[Lan et al., 2015]	68.0	65.4	89.1
[Yue-Hei Ng et al., 2015]	–	–	88.6
[Simonyan and Zisserman, 2014]	–	59.4	88.0
[Hoai and Zisserman, 2014]	73.6	60.8	–
[Peng et al., 2014]	–	66.8	–
[Wu et al., 2014]	–	56.4	84.2
[Jain et al., 2013]	62.5	52.1	–
[Wang and Schmid, 2013]	64.3	57.2	–
[Wang et al., 2013]	58.2	46.6	–
[Taylor et al., 2010]	46.6	–	–

Table 4 : Comparison with the state-of-the-art methods.

Parameter Evaluation : window size and stride

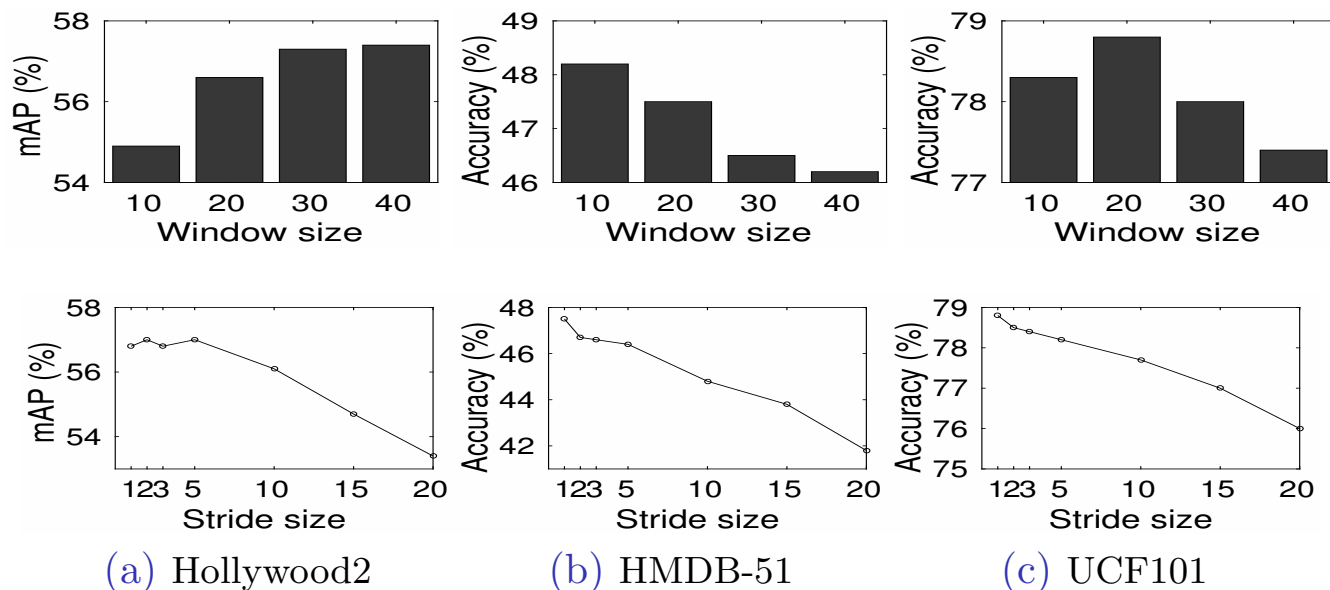


Figure 4 : Activity recognition performance versus window size (top) and stride (bottom).

Parameter Evaluation : hierarchy depth

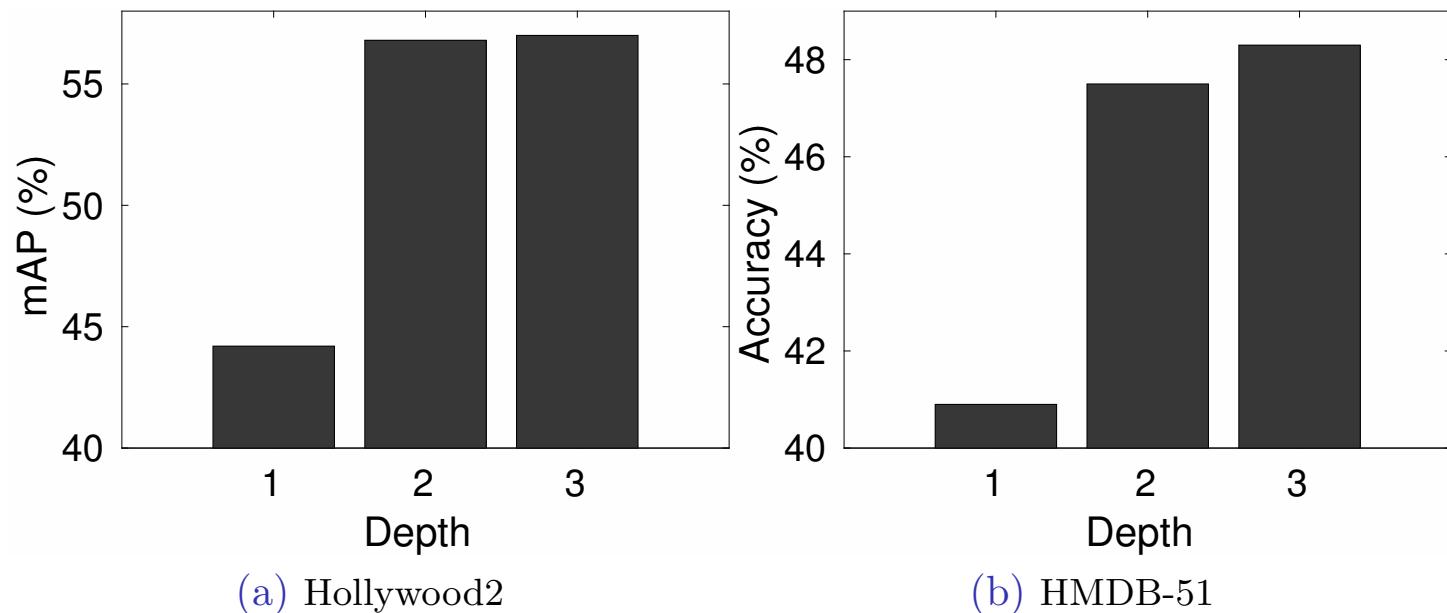


Figure 5 : Activity recognition performance versus hierarchy depth on Hollywood2 and HMDB-51.

PART IV. Discriminative dynamics learning with CNN

Discriminative dynamics learning with CNN - ICML 2016

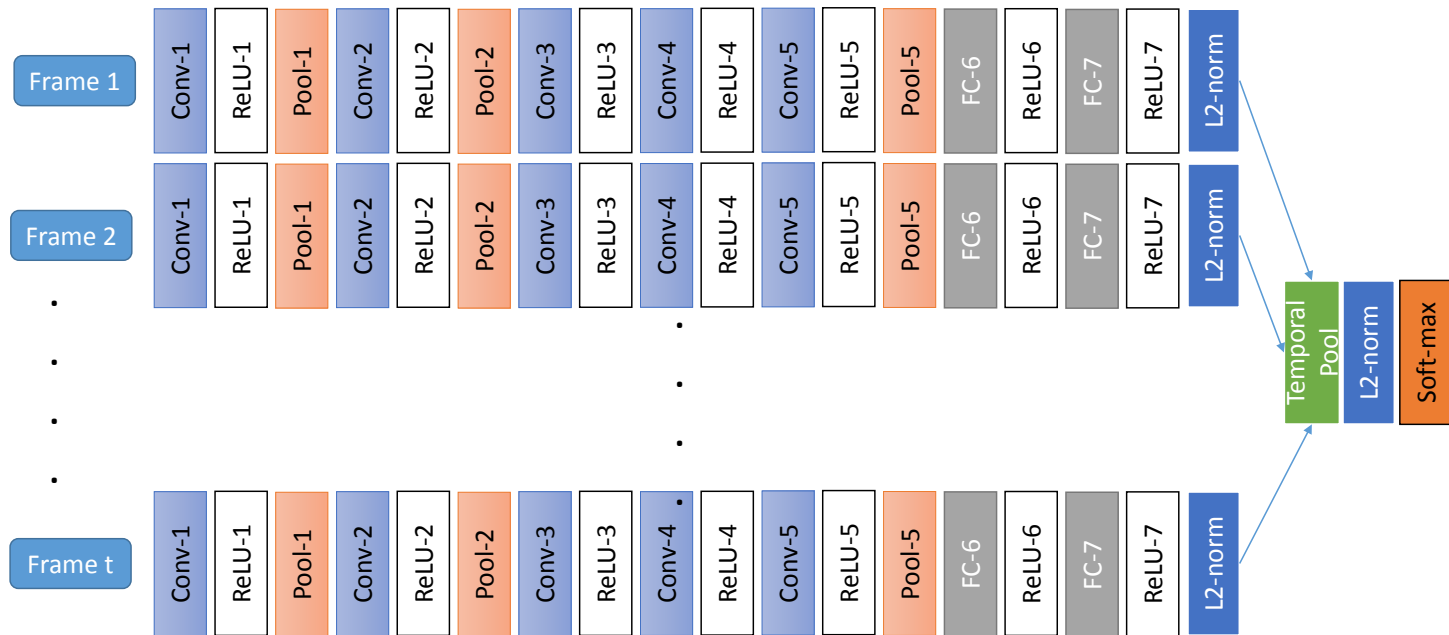


Figure 6 : The CNN network architecture takes a sequence of frames from a video as inputs and feed forward till the end of the temporal pooling layer. At the temporal polling layer, the sequence of vectors are encoded by

General learning framework

- $\vec{\mathbf{x}}$ is the sequence of frames.
- ψ_{θ} is the learn-able feature function (e.g. CNN)
- ϕ is the sequence encoding function or any leaning program that returns an encoding.
- h_{β} is a classifier

$$\vec{\mathbf{x}} = \langle \mathbf{x}_t \rangle \xrightarrow{\psi_{\theta}} \langle \mathbf{v}_t \rangle \xrightarrow{\phi} \mathbf{u} \xrightarrow{h_{\beta}} y \quad (8)$$

Given a dataset of sequence-label pairs, $\{(\vec{\mathbf{x}}^{(i)}, y^{(i)})\}_{i=1}^n$, our goal is to learn all the parameters.

Let $\Delta(\cdot, \cdot)$ be a loss function.

$$\Delta(y, h_{\beta}(\mathbf{u})) = -\log P(y \mid \vec{\mathbf{x}}). \quad (9)$$

Temporal encoding of the sequence $\langle \mathbf{v}_t \rangle$

The sequence of CNN vectors is given by $\vec{\mathbf{v}} = \langle \mathbf{v}_1, \dots, \mathbf{v}_T \rangle$

The temporal encoding is obtained by

$$\mathbf{u} = \phi(\vec{\mathbf{v}}) \in \mathbb{R}^q. \quad (10)$$

To get the encoding we need to solve the following optimization

$$\mathbf{u} \in \arg \min_{\mathbf{u}'} f(\vec{\mathbf{v}}, \mathbf{u}') \quad (11)$$

For mean pooling the sequence encoder is given by

$$\text{avg}(\vec{\mathbf{v}}) = \underset{\mathbf{u}}{\text{argmin}} \left\{ \frac{1}{2} \sum_{t=1}^T \|\mathbf{u} - \mathbf{v}_t\|^2 \right\}. \quad (12)$$

We jointly estimate the parameters of the feature function and prediction function by minimizing the regularized empirical risk. Our learning problem is

$$\begin{aligned} & \text{minimize}_{\boldsymbol{\theta}, \boldsymbol{\beta}} && \sum_{i=1}^n \Delta(y^{(i)}, h_{\boldsymbol{\beta}}(\mathbf{u}^{(i)})) + R(\boldsymbol{\theta}, \boldsymbol{\beta}) \\ & \text{subject to} && \mathbf{u}^{(i)} \in \operatorname{argmin}_{\mathbf{u}} f(\vec{\mathbf{v}}^{(i)}, \mathbf{u}) \end{aligned} \quad (13)$$

Lemma

Let $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous function with first and second derivatives. Let

$$\mathbf{g}(x) = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^n} f(x, \mathbf{y})$$

, then

$$\mathbf{g}'(x) = -f_{YY}(x, \mathbf{g}(x))^{-1} f_{XY}(x, \mathbf{g}(x)).$$

where $f_{YY} \doteq \nabla_{\mathbf{y}\mathbf{y}}^2 f(x, \mathbf{y}) \in \mathbb{R}^{n \times n}$ and $f_{XY} \doteq \frac{\partial}{\partial x} \nabla_{\mathbf{y}} f(x, \mathbf{y}) \in \mathbb{R}^n$.

Gradient computation of sequence encoder

$$f(\theta, \mathbf{u}) = \frac{1}{2} \|\mathbf{u}\|^2 + \frac{C}{2} \sum_{t=1}^T \left[|t - \mathbf{u}^T \mathbf{v}_t| - \epsilon \right]_{\geq 0}^2 \quad (14)$$

Now compute the gradients of the sequence encoder $\frac{d}{d\theta} \operatorname{argmin}_{\mathbf{u}} f(\theta, \mathbf{u})$ using lemma.

$$\frac{d}{d\theta} \operatorname{argmin}_{\mathbf{u}} f(\theta, \mathbf{u}) = \begin{pmatrix} I + C \sum_{e_t \neq 0} \mathbf{v}_t \mathbf{v}_t^T \\ C \sum_{e_t \neq 0} e_t \psi'_\theta(\mathbf{x}_t) - \mathbf{u}^T \psi'_\theta(\mathbf{x}_t) \mathbf{v}_t \end{pmatrix}^{-1} \quad (15)$$

Results of End-to-end learning of video representations

Table 5 : Classification performance in average precision for activity recognition on the Hollywood2 dataset.




CLASS	MEAN	MAX	RANKPOOL	MEAN	MAX	RANKPOOL
	SVM			CNN		
ANSWERPHONE	23.6	19.5	35.3	29.9	28.0	25.0
DRIVECAR	60.9	50.8	40.6	55.6	48.6	56.9
EAT	19.7	22.0	16.7	27.8	22.0	24.2
FIGHTPERSON	45.6	28.3	28.1	26.6	17.6	30.4
GETOUTCAR	39.5	29.2	28.1	48.9	43.8	55.5
HANDSHAKE	28.3	24.4	34.2	38.4	40.0	32.0
HUGPERSON	30.2	23.9	22.1	25.9	26.6	33.2
KISS	38.2	27.5	36.8	50.6	45.7	54.2
RUN	55.2	53.0	39.4	59.6	52.5	61.0
SITDOWN	30.0	28.8	32.1	30.6	30.0	39.6
SITUP	23.0	20.2	18.7	23.8	26.4	25.4
STANDUP	34.6	32.4	39.9	37.4	34.8	49.9
AVG	35.7	30.0	31.0	37.9	34.7	40.6

- LEAR - Improved Trajectories Video Description - Heng Wang
lear.inrialpes.fr/people/wang/improved_trajectories
- Fisher encoding VLFeat <http://www.vlfeat.org/index.html>
- Rank pooling code
<https://bitbucket.org/bfernando/videodarwin>
- Dynamic Image Nets
<https://github.com/hbilen/dynamic-image-nets>





End

- Thank you!





References I

-  Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., and Tuytelaars, T. (2015).
Modeling video evolution for action recognition.
In *CVPR*.
-  Hoai, M. and Zisserman, A. (2014).
Improving human action recognition using score distribution and ranking.
In *ACCV*.
-  Jain, M., Jégou, H., and Bouthemy, P. (2013).
Better exploiting motion for better action recognition.
In *CVPR*.





References II




-  Ji, S., Xu, W., Yang, M., and Yu, K. (2013).
3d convolutional neural networks for human action recognition.
Pattern Analysis and Machine Intelligence, IEEE Transactions on,
35(1):221–231.
-  Lan, Z., Lin, M., Li, X., Hauptmann, A. G., and Raj, B. (2015).
Beyond gaussian pyramid: Multi-skip feature stacking for action
recognition.
In *CVPR*.
-  Laptev, I. (2005).
On space-time interest points.
IJCV, 64:107–123.
-  Peng, X., Zou, C., Qiao, Y., and Peng, Q. (2014).
Action recognition with stacked fisher vectors.
In *ECCV*.



References III

-  Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010). Large-scale image retrieval with compressed fisher vectors. In *CVPR*.
-  Pfister, T., Charles, J., and Zisserman, A. (2014). Domain-adaptive discriminative one-shot learning of gestures. In *ECCV*.
-  Rohrbach, M., Amin, S., Andriluka, M., and Schiele, B. (2012). A database for fine grained activity detection of cooking activities. In *CVPR*.
-  Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576.

References IV

-  Taylor, G. W., Fergus, R., LeCun, Y., and Bregler, C. (2010). Convolutional learning of spatio-temporal features. In *ECCV*.
-  Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103:60–79.
-  Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *ICCV*.
-  Wu, J., Cheng, J., Zhao, C., and Lu, H. (2013). Fusing multi-modal features for gesture recognition. In *ICMI*.

-  Wu, J., Zhang, Y., and Lin, W. (2014).
Towards good practices for action video encoding.
In *CVPR*.
-  Yao, A., Van Gool, L., and Kohli, P. (2014).
Gesture recognition portfolios for personalization.
In *CVPR*.
-  Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015).
Beyond short snippets: Deep networks for video classification.
In *CVPR*.

-  Zha, S., Luisier, F., Andrews, W., Srivastava, N., and Salakhutdinov, R. (2015).
Exploiting image-trained CNN architectures for unconstrained video classification.
In *BMVC*.
-  Zhou, Y., Ni, B., Yan, S., Moulin, P., and Tian, Q. (2014).
Pipelining localized semantic features for fine-grained action recognition.
In *ECCV*.