



*Orange Labs - Rennes, France*  
*Multimedia Contents Analysis Technologies*  
[www.francetelecom.com/rd/](http://www.francetelecom.com/rd/)



*LIRIS, CNRS / INSA Lyon - Lyon, France*  
*Feature Extraction and Identification*  
<http://liris.cnrs.fr>

# Sequential Deep Learning for Human Action Recognition

Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia and Atilla Baskurt

November 16<sup>th</sup> 2011

- Action recognition for modeling human behavior.
- State of the art methods for action recognition rely mostly on **hand-crafted** features (Harris-3D, MoSIFT...).
- **Deep models** are particular learning machines that automatically extract features, based only on training examples.
- Deep models are well known for their capabilities to automatically extract features in 1D and 2D tasks, based on learning examples.
- 2D+t (or 3D) extension is still an open issue, and few attempts operate on short sub-sequences.
- Using deep neural networks to automatically learn to extract features and classify entire sequences based on the evolution of these features.

- Provide a complete recognition scheme that operates on entire sequences.
- Investigate the extension to 3D of a widely used deep architecture, LeCun's **ConvNets**, to automatically learn space-time features.
- Take advantage of sequential learning machines, namely **recurrent neural networks**, to exploit the evolution of learned features over time.
- Evaluate and compare our method on the standard **KTH human actions** dataset (containing 6 actions).

Automated Space-Time Feature Construction with 3D ConvNets

Sequence Labeling using Learned Features

Experimental results on KTH Dataset

Automated Space-Time Feature Construction with 3D ConvNets

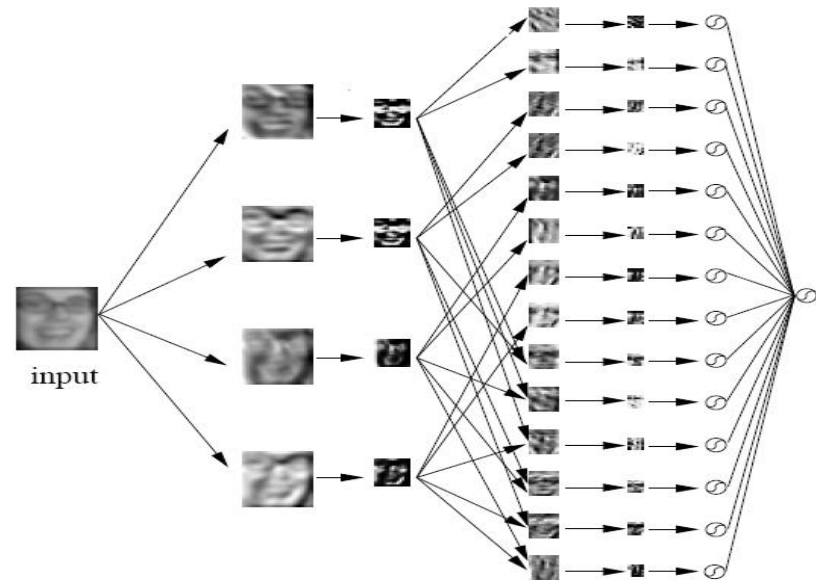
Sequence Labeling using Learned Features

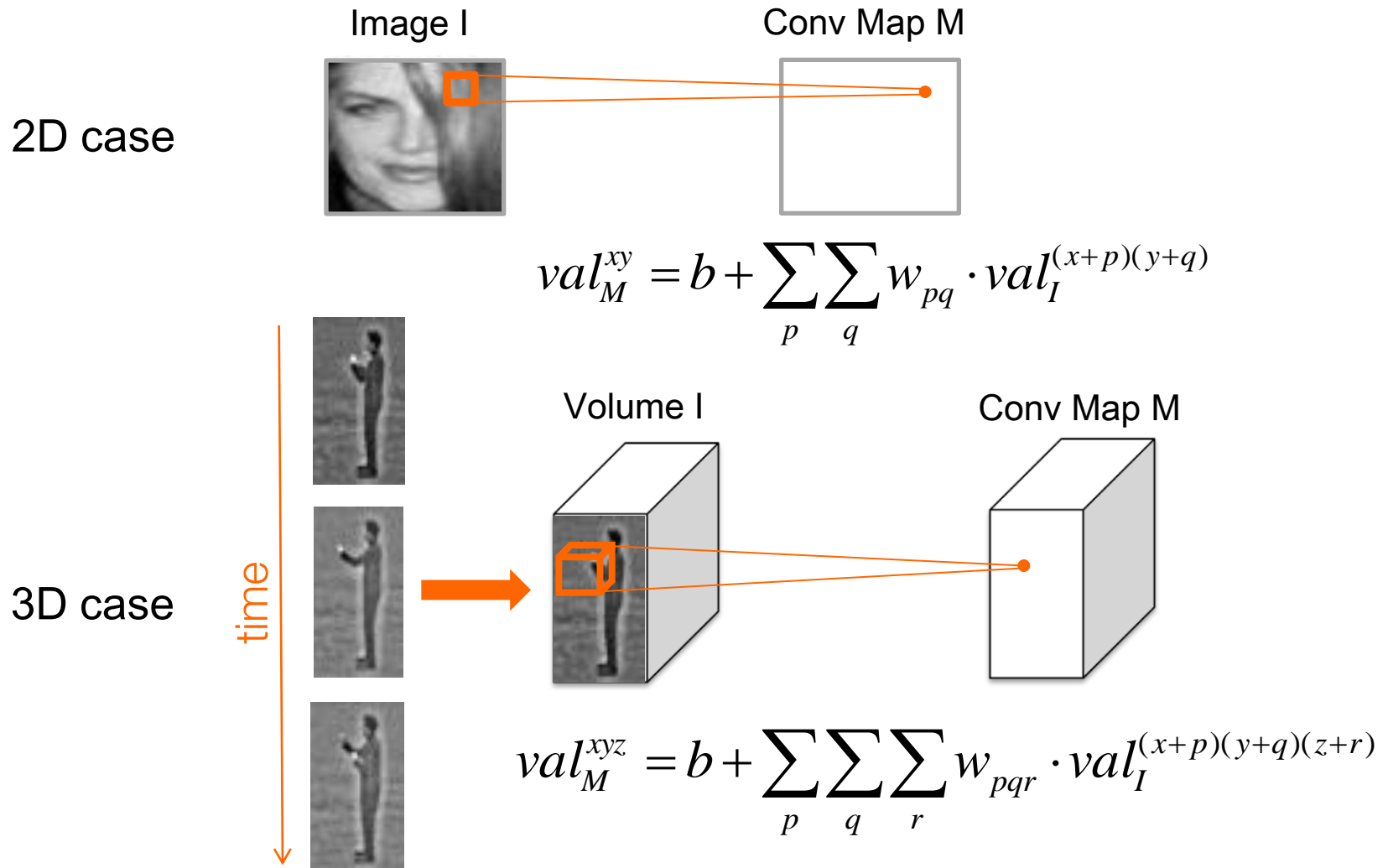
Experimental results on KTH Dataset

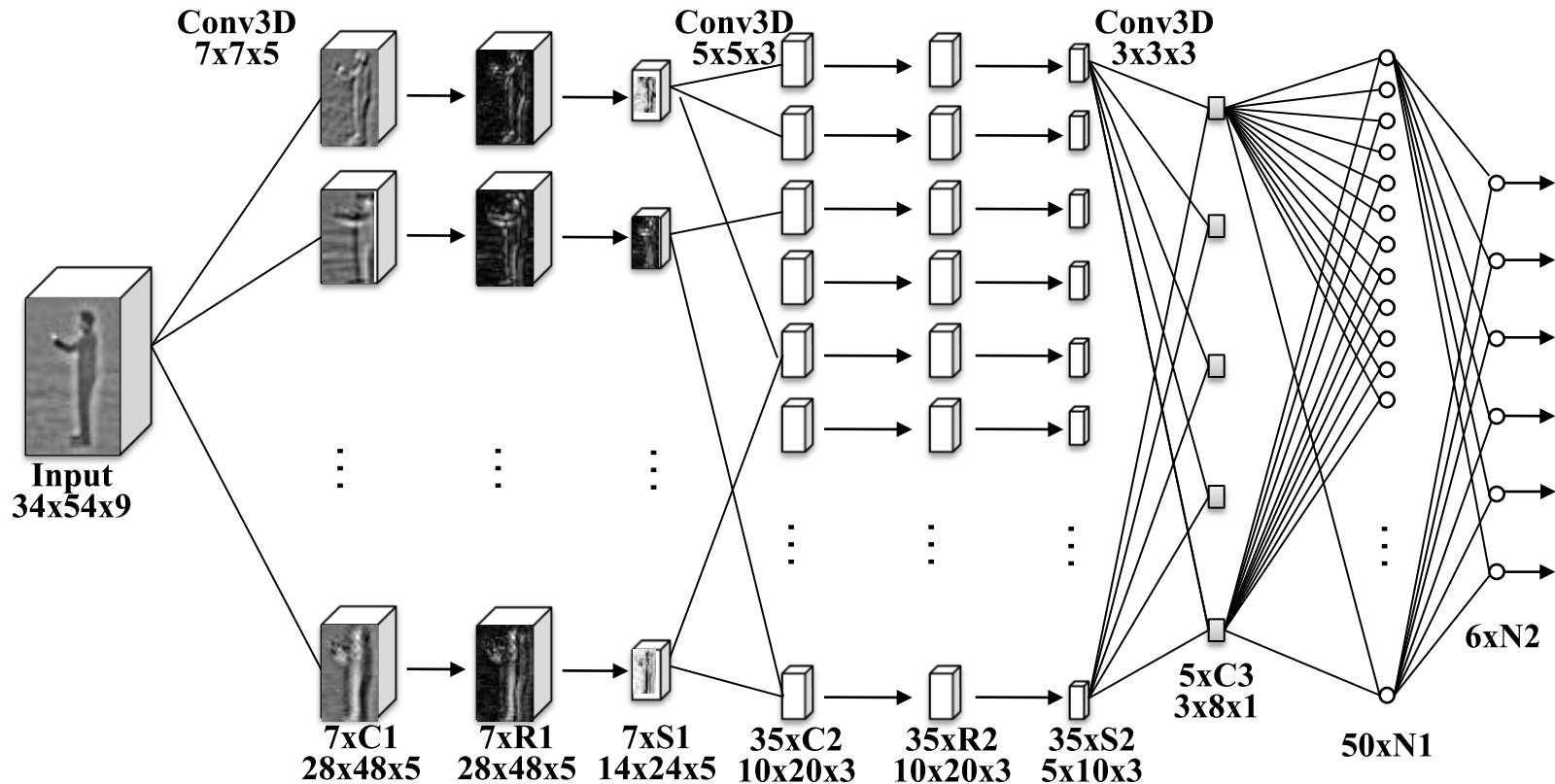
- LeCun's ConvNets are biologically-inspired deep models especially designed to handle 2D images.
- Can be seen as a succession of:
  - 2D convolutions: to capture spatial salient information.
  - Sub-samplings: to reduce dimension.
  - Rectifications: a non-linearity layer using abs. value (optional).

- Example of application:

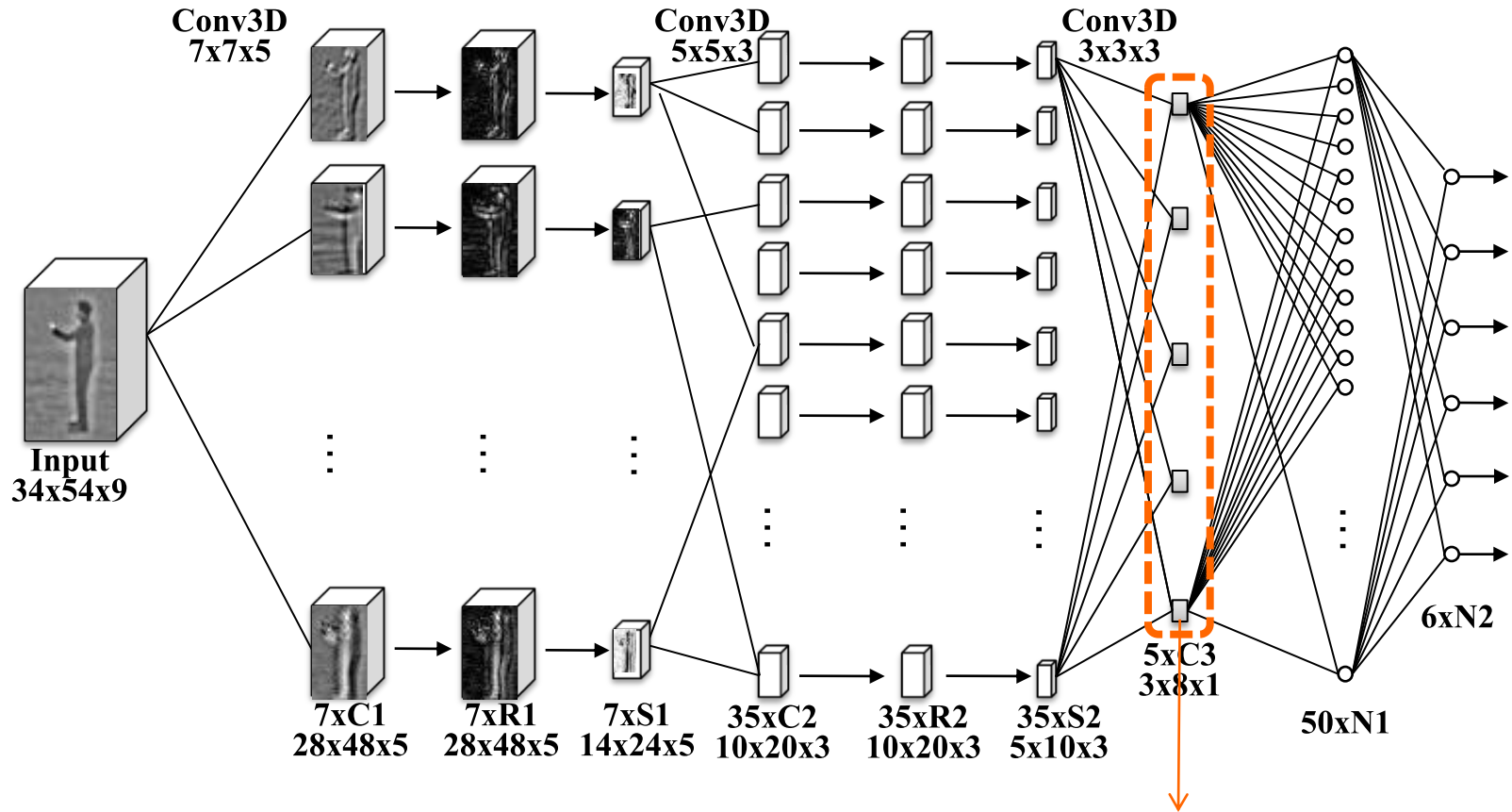
Convolutional Face Finder  
*[Garcia & Delakis, PAMI 2004]*







- The network is trained with standard *online backpropagation*, *targetting action class of the video sub-sequence*.



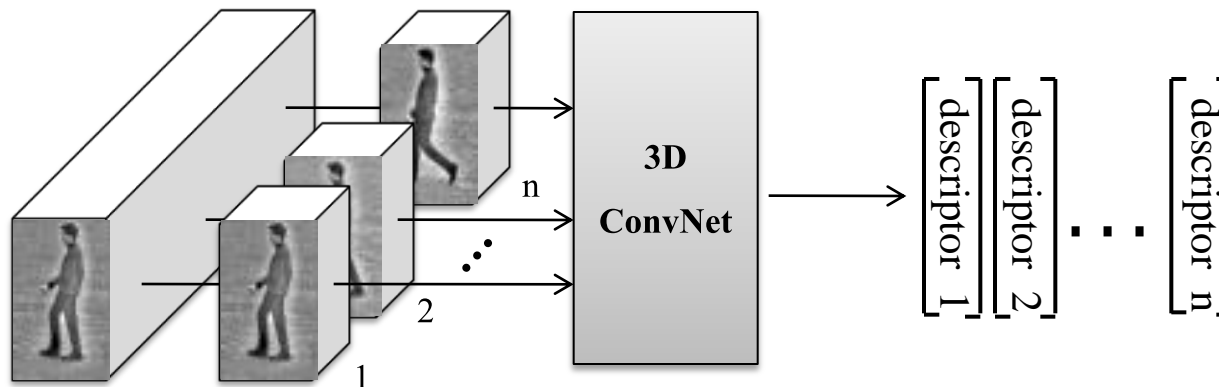
- The network is trained with standard *online backpropagation*, targeting *action class of the video sub-sequence*.

Automated Space-Time Feature Construction with 3D ConvNets

Sequence Labeling using Learned Features

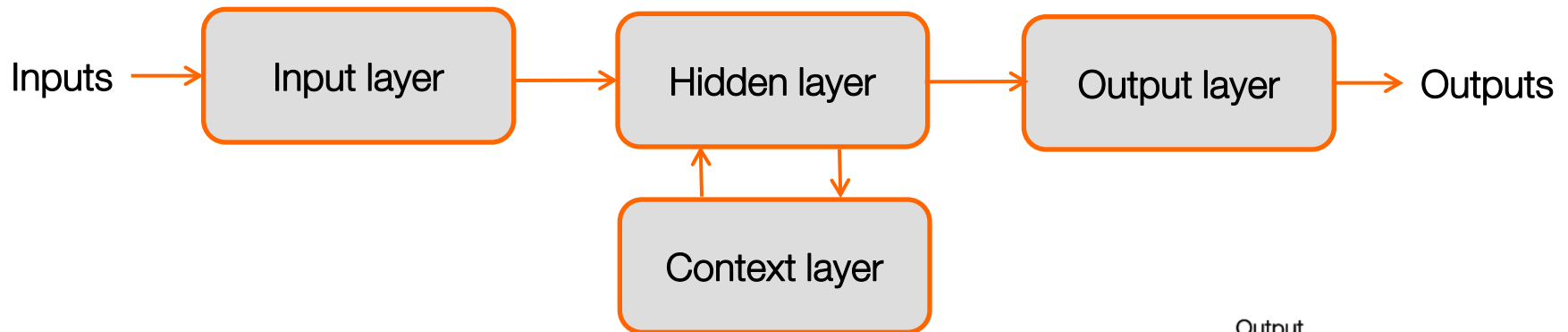
Experimental results on KTH Dataset

- Each C3 descriptor corresponds to a short sub-sequence (9 frames) from the entire video.

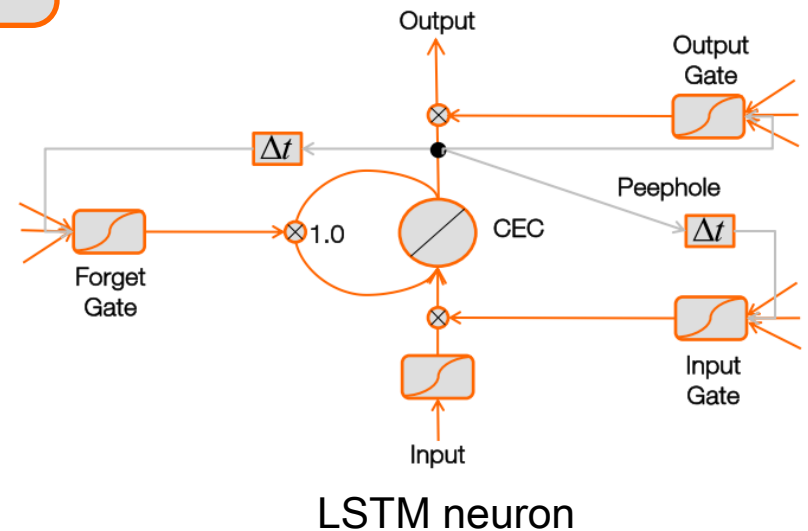


- A video is represented as a sequence of descriptors corresponding to 3D-ConvNets learned features.
- These sequences are used as input to train a recurrent neural network.

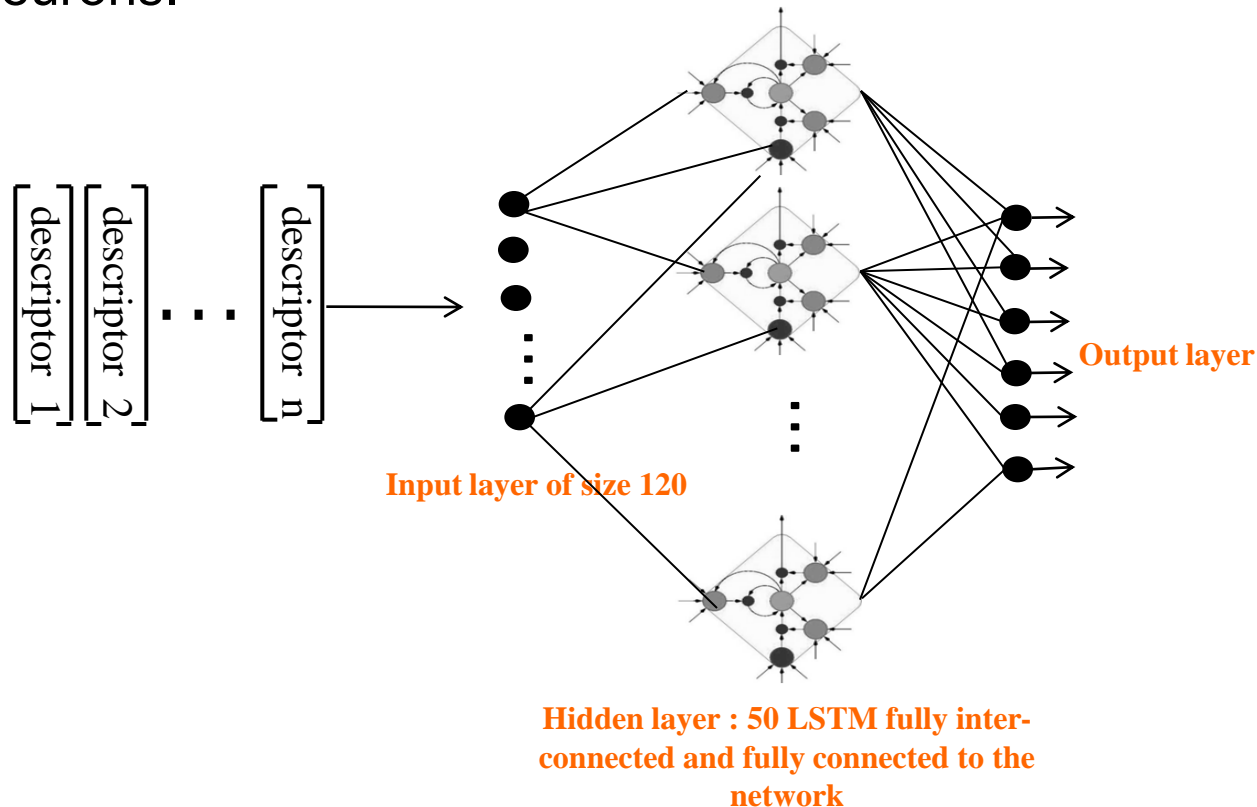
- RNN = ANN with a **context layer** which allows to remember and use previous inputs.



- Schmidhuber et al. introduced a particular recurrent architecture, which outperforms classical RNNs: the **Long Short-Term Memory**:



- Descriptors sequences are used to train a RNN with LSTM neurons.

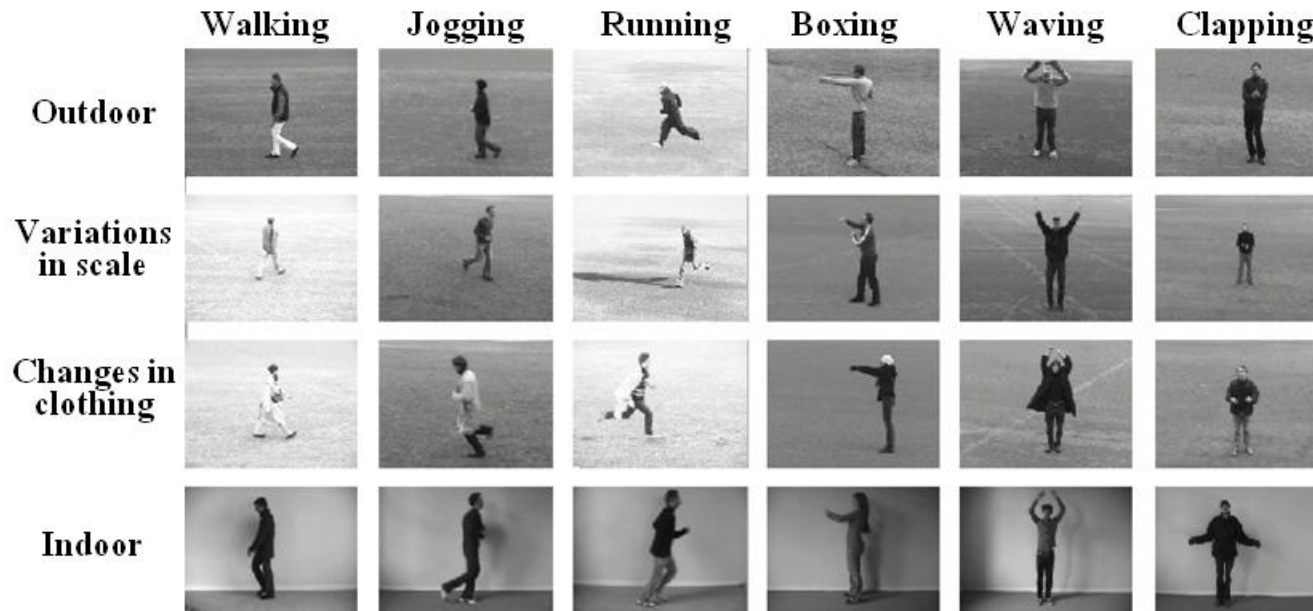


- The network is trained with *online backpropagation through time* algorithm, targeting action class at each timestep.

Automated Space-Time Feature Construction with 3D ConvNets

Sequence Labeling using Learned Features

Experimental results on KTH Dataset






- 25 persons performing 6 actions in 4 scenarios.
- Two versions used in the literature:
  - KTH1: 599 long sequences (several iterations of the same action / video)
  - KTH2: 2391 short sequences (one iteration of the same action / video)

- Evaluation protocol:
  - 5 randomly selected training/test configurations.
  - For each configuration:
    - 16 randomly selected persons for training.
    - the 9 others for test.
- We have also tested two other approaches:
  - 3D-ConvNets + majority voting:
    - to evaluate the LSTM contribution.
  - Harris3D + LSTM:
    - to evaluate the learned features relevance vs hand-crafted ones.

Dataset	KTH1		KTH2		
Method	3D ConvNets + voting	3D ConvNets + LSTM	Harris 3D + LSTM	3D ConvNets + voting	3D ConvNets + LSTM
Average accuracy	91.04	<b>94.39</b>	87.78	89.40	<b>92.17</b>

Dataset	Evaluation Protocol	Method	Accuracy
KTH1	Cross validation with 5 runs	<b>Our method</b>	<b>94.39</b>
		<b>Jhuang et al.</b>	<b>91.70</b>
		Gao et al.	95.04
		Schindler et al.	92.70
	Leave-one-out	Gao et al.	96.33
		Chen et al.	95.83
KTH2	Cross validation with 5 runs	<b>Our method</b>	<b>92.17</b>
		<b>Ji et al.</b>	<b>90.20</b>
		Gao et al.	93.57
	Leave-one-out	<b>Taylor et al.</b>	<b>90.00</b>
		Kim et al.	95.33

-  our method
-  other deep models
-  hand-crafted features

- A 2-steps neural scheme which learns to automatically recognize actions relying only on training examples, without any prior knowledge.
- Despite its fully automated nature, the proposed model gives competitive results, among the best of related work on KTH dataset.
- The automatically learned features are more relevant for this task than the hand-crafted Harris-3D ones.
- Future work: verify the genericity of our approach by testing it on recent more challenging datasets.

Thank you for your attention

Moez Baccouche

<http://liris.cnrs.fr/moez.baccouche/>

[moez.baccouche@orange.com](mailto:moez.baccouche@orange.com)