Sequential Deep Learning for Human Action Recognition

Moez Baccouche[1,2], Franck Mamalet[1],
Christian Wolf[2], Christophe Garcia[2], and Atilla Baskurt[2]

[1] Orange Labs, 4 rue du Clos Courtel, 35510 Cesson-Sévigné, France
firstname.surname@orange-ftgroup.com
[2] LIRIS, UMR 5205 CNRS, INSA-Lyon, F-69621, France
firstname.surname@liris.cnrs.fr

Abstract. We propose in this paper a fully automated deep model, which learns to classify human actions without using any prior knowledge. The first step of our scheme, based on the extension of *Convolutional Neural Networks* to 3D, automatically learns spatio-temporal features. A *Recurrent Neural Network* is then trained to classify each sequence considering the temporal evolution of the learned features for each timestep. Experimental results on the KTH dataset show that the proposed approach outperforms existing deep models, and gives comparable results with the best related works.

Keywords: Human action recognition, deep models, 3D convolutional neural networks, long short-term memory, KTH human actions dataset.

1 Introduction and Related Work

Automatic understanding of human behaviour and its interaction with his environment have been an active research area in the last years due to its potential application in a variety of domains. To achieve such a challenging task, several research fields focus on modeling human behaviour under its multiple facets (emotions, relational attitudes, actions, etc.). In this context, recognizing the behaviour of a person appears to be crucial when interpreting complex actions. Thus, a great interest has been granted to human action recognition, especially in real-world environments.

Among the most popular state-of-the-art methods for human action recognition, we can mention those proposed by Laptev et al. [13], Dollar et al. [3] and others [12,17,2,4], which all use engineered motion and texture descriptors calculated around spatio-temporal interest points, which are manually engineered. The *Harris-3D* detector [13] and the *Cuboid* detector [3] are likely the most used space-time salient points detectors in the literature. Nevertheless, even if their extraction process is fully automated, these so-called *hand-crafted* features are especially designed to be optimal for a specific task. Thus, despite their high performances, these approaches main drawback is that they are highly problem dependent.

A.A. Salah and B. Lepri (Eds.): HBU 2011, LNCS 7065, pp. 29–39, 2011.

In last years, there has been a growing interest in approaches, so-called *deep models*, that can learn multiple layers of feature hierarchies and automatically build high-level representations of the raw input. They are thereby more generic since the feature construction process is fully automated. One of the most used deep models is the *Convolutional Neural Network* architecture [14,15], hereafter ConvNets, which is a bioinspired hierarchical multilayered neural network able to learn visual patterns directly from the image pixels without any pre-processing step. If ConvNets were shown to yield very competitive performances in many image processing tasks, their extension to the video case is still an open issue, and, so far, the few attempts either make no use of the motion information [20], or operate on hand-crafted inputs (spatio-temporal outer boundaries volume in [11] or hand-wired combination of multiple input channels in [10]). In addition, since these models take as input a small number of consecutive frames (typically less than 15), they are trained to assign a vector of features (and a label) to short sub-sequences and not to the entire sequence. Thus, even if the learned features, taken individually, contains temporal information, their evolution over time is completely ignored. Though, we have shown in our previous work [1] that such information does help discriminating between actions, and is particularly usable by a category of learning machines, adapted to sequential data, namely *Long Short-Term Memory* recurrent neural networks (LSTM) [6].

In this paper, we propose a two-steps neural-based deep model for human action recognition. The first part of the model, based on the extension of ConvNets to 3D case, automatically learns spatio-temporal features. Then, the second step consists in using these learned features to train a recurrent neural network model in order to classify the entire sequence. We evaluate the performances on the KTH dataset [24], taking particular care to follow the evaluation protocol recommendations discussed in [4]. We show that, without using the LSTM classifier, we obtain comparable results with other deep models based approaches [9,26,10]. We also demonstrate that the introduction of the LSTM classification leads to significant performance improvement, reaching average accuracies among the best related results.

The rest of the paper is organized as follows. Section 2 outlines some ConvNets fundamentals and the feature learning process. We present in Section 3 the recurrent neural scheme for entire sequence labelling. Finally, experimental results, carried out on the KTH dataset, will be presented in Section 4.

2 Deep Learning of Spatio-Temporal Features

In this section, we describe the first part of our neural recognition scheme. We first present some fundamentals of 2D-ConvNets, and then discuss their extension in 3D and describe the proposed architecture.

2.1 Convolutional Neural Networks (ConvNets)

Despite their generic nature, deep models were not used in many applications until the late nineties because of their inability to treat "real world" data.

Indeed, early deep architectures dealt only with 1-D data or small 2D-patches. The main problem was that the input was "fully connected" to the model, and thus the number of free parameters was directly related to the input dimension, making these approaches inappropriate to handle "pictoral" inputs (natural images, videos...).

Therefore, the convolutional architecture was introduced by LeCun et al. [14,15] to alleviate this problem. ConvNets are the adaptation of multilayered neural deep architectures to deal with real world data. This is done by the use of local *receptive fields* whose parameters are forced to be identical for all its possible locations, a principle called *weight sharing*. Schematically, LeCun's ConvNet architecture [14,15] is a succession of layers alternating 2D-convolutions (to capture salient information) and sub-samplings (to reduce dimension), both with trainable weights. Jarret et al. [8] have recommended the use of rectification layers (which simply apply absolute value to its input) after each convolution, which was shown to significantly improve performances, when input data is normalized.

In the next sub-section, we examine the adaptation of ConvNets to video processing, and describe the 3D-ConvNets architecture that we used in our experiments on the KTH dataset.

2.2 Automated Space-Time Feature Construction with 3D-ConvNets

The extension from 2D to 3D in terms of architecture is straightforward since 2D convolutions are simply replaced by 3D ones, to handle video inputs. Our proposed architecture, illustrated in Figure 1, also uses 3D convolutions, but is different from [11] and [10] in the fact that it uses only raw inputs.

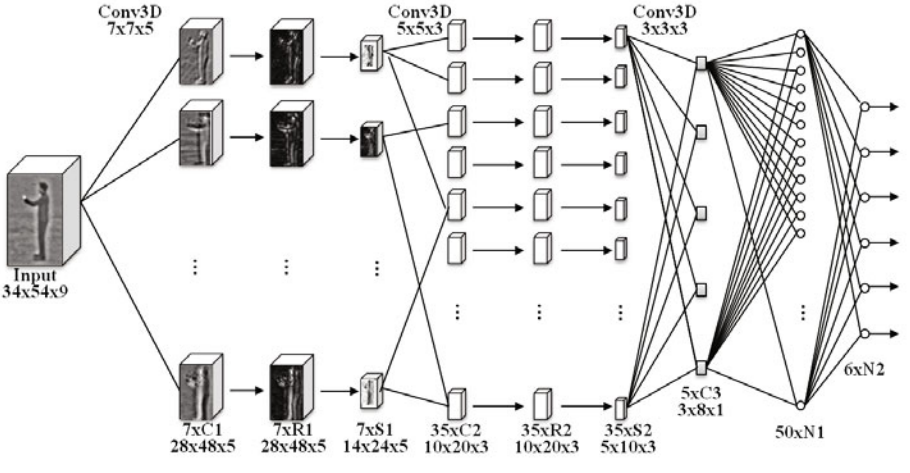

Fig. 1. Our 3D-ConvNet architecture for spatio-temporal features construction

This architecture consists of 10 layers including the input. There are two alternating convolutional, rectification and sub-sampling layers C1, R1, S1 and C2, R2, S2 followed by a third convolution layer C3 and two neuron layers N1 and N2. The size of the 3D input layer is $34 \times 54 \times 9$, corresponding to 9 successive frames of 34×54 pixels each. Layer C1 is composed of 7 feature maps of size $28 \times 48 \times 5$ pixels. Each unit in each feature map is connected to a 3D $7 \times 7 \times 5$ neighborhood into the input retina. Layer R1 is composed of 7 feature maps, each connected to one feature map in C1, and simply applies absolute value to its input. Layer S1 is composed of 7 feature maps of size $14 \times 24 \times 5$, each connected to one feature map in R1. S1 performs sub-sampling at a factor of 2 in spatial domain, aiming to build robustness to small spatial distortions. The connection scheme between layers S1 and C2 follows the same principle described in [5], so that, C2 layer has 35 feature maps performing $5 \times 5 \times 3$ convolutions. Layers R2 and S2 follow the same principle described above for R1 and S1. Finally, layer C3 consists of 5 feature maps fully-connected to S2 and performing $3 \times 3 \times 3$ convolutions. At this stage, each C3 feature map contains $3 \times 8 \times 1$ values, and thus, the input information is encoded in a vector of size 120. This vector can be interpreted as a descriptor of the salient spatio-temporal information extracted from the input. Finally, layers N1 and N2 contain a classical multilayer perceptron with one neuron per action in the output layer. This architecture corresponds to a total of 17,169 trainable parameters (which is about 15 times less than the architecture used in [10]). To train this model, we used the algorithm proposed in [14], which is the standard *online Backpropagation with momentum* algorithm, adapted *to weight sharing.*

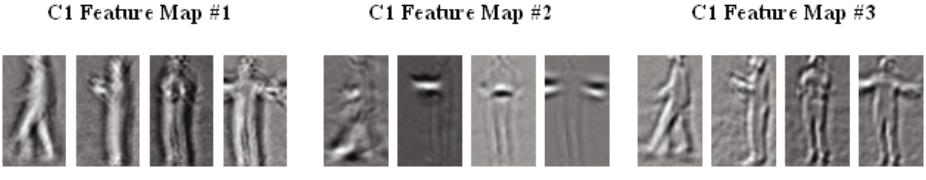

Fig. 2. A subset of 3 automatically constructed C1 feature maps (of 7 total), each one corresponding, from left to right, to the actions walking, boxing, hand-claping and hand-waving from the KTH dataset

Once the 3D-ConvNet is trained on KTH actions, and since the spatio-temporal feature construction process is fully automated, it's interesting to examine if the learned features are visually interpretable. We report in Figure 2 a subset of learned C1 feature maps, corresponding each to some actions from the KTH dataset. Even if finding a direct link with engineered features is not straightforward (and not necessarily required) the learned feature maps seem to capture visually relevant information (person/background segmentation, limbs involved during the action, edge information...).

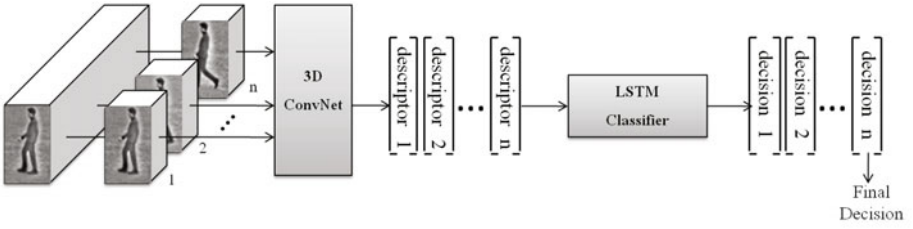

Fig. 3. An overview of our two-steps neural recognition scheme

In the next section, we describe how these features are used to feed a recurrent neural network classifier, which is trained to recognize the actions based on the temporal evolution of features.

3 Sequence Labelling Considering the Temporal Evolution of Learned Features

Once the features are automatically constructed with the 3D-ConvNet architecture as described in Section 2, we propose to learn to label the entire sequence based on the accumulation of several individual decisions corresponding each to a small temporal neighbourhood which was involved during the 3D-ConvNets learning process (see Figure 3). This allows to take advantage of the temporal evolution of the features, in comparison with the majority voting process on the individual decisions.

Among state of the art learning machines, Recurrent Neural Networks (RNN) are one of the most used for temporal analysis of data, because of their ability to take into account the context using recurrent connections in the hidden layers. It has been demonstrated in [6] that if RNN are able to learn tasks which involve short time lags between inputs and corresponding teacher signals, this *short-term* memory becomes insufficient when dealing with "real world" sequence processing, e.g video sequences. In order to alleviate this problem, Schmidhuber et al. [6] have proposed a specific recurrent architecture, namely *Long Short-Term Memory* (LSTM). These networks use a special node called *Constant Error Carousel* (CEC), that allows for constant error signal propagation through time. The second key idea in LSTM is the use of multiplicative gates to control the access to the CEC. We have shown in our previous work [1] that LSTM are efficient to label sequences of descriptors corresponding to hand-crafted features.

In order to classify the action sequences, we propose to use a *Recurrent Neural Network* architecture with one hidden layer of LSTM cells. The input layer of this RNN consists in 120 C3 output values per time step. LSTM cells are fully connected to these inputs and have also recurrent connexions with all the LSTM cells. Output layer consists in neurons connected to LSTM outputs at each time step. We have tested several network configuration, varying the number of hidden LSTM. A configuration of 50 LSTM was found to be a good compromise for

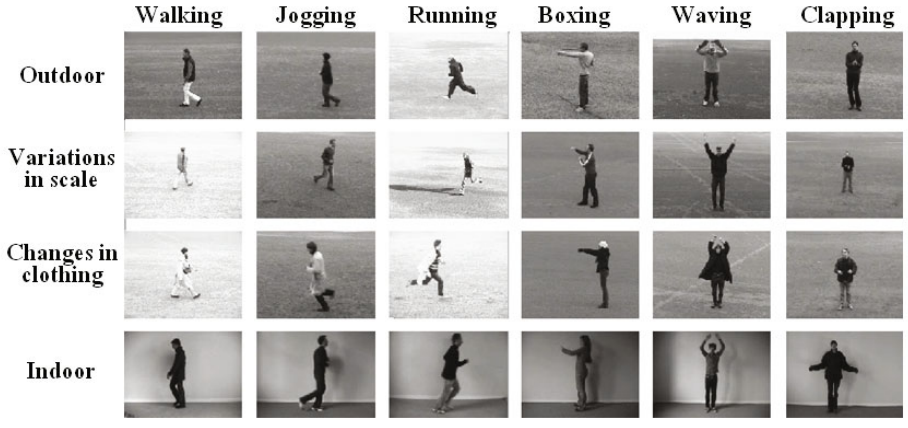

Fig. 4. A sample of actions/scenarios from the KTH dataset [24]

this classification task. This architecture corresponds to about 25,000 trainable parameters. The network was trained with *online backpropagation through time with momentum* [6].

4 Experiments on KTH Dataset

The KTH dataset was provided by Schuldt et al. [24] in 2004 and is the most commonly used public human actions dataset. It contains 6 types of actions (walking, jogging, running,boxing, hand-waving and hand-clapping) performed by 25 subjects in 4 different scenarios including indoor, outdoor, changes in clothing and variations in scale (see Figure 4). The image size is of 160×120 pixels, and temporal resolution is of 25 frames per second. There are considerable variations in duration and viewpoint. All sequences were taken over homogeneous backgrounds, but hard shadows are present.

As in [4], we rename the KTH dataset in two ways: the first one (the original one) where each person performs the same action 3 or 4 times in the same video, is named KTH1 and contains 599 long sequences (with a length between 8 and 59 seconds) with several "empty" frames between action iterations. The second, named KTH2, is obtained by splitting videos in smaller ones where a person does an action only one time, and contains 2391 sequences (with a length between 1 and 14 seconds).

4.1 Evaluation Protocol

In [4], Gao et al. presented a comprehensive study on the influence of the evaluation protocol on the final results. It was shown that the use of different experimental configurations can lead to performance differences up to 9%.

Furthermore, authors demonstrated that the same method, when evaluated on KTH1 or KTH2 can have over 5.85% performance deviations. Action recognition methods are usually directly compared although they use different testing protocols or/and datasets (KTH1 or KTH2), which distorts the conclusions. In this paper, we choose to evaluate our method using cross-validation, in which 16 randomly-selected persons are used for training, and the other 9 for testing. Recognition performance corresponds to the average across 5 trials. Evaluations are performed on both KTH1 and KTH2.

4.2 Experimental Results

The two-steps model was trained as described above. Original videos underwent the following steps: spatial down-sampling by a factor of 2 horizontally and vertically to reduce the memory requirement, extracting the person-centred bounding box as in [9,10], and applying 3D *Local Contrast Normalization* on a $7 \times 7 \times 7$ neighbourhood, as recommended in [8]. Note that we do not use any complex pre-processing (optical flow, gradients, motion history...). We also generated vertically flipped and mirrored versions of each training sample to increase the number of examples. In our experiments, we observed that, both for 3D-ConvNets and LSTM, no overtraining is observed without any validation sequence and stopping when performances on training set no longer rise. Obtained results, corresponding to 5 randomly selected training/test configurations are reported on Table 1.

Table 1. Summary of experimental results using 5 randomly selected configurations from KTH1 and KTH2

		Config.1	Config.2	Config.3	Config.4	Config.5	Average
KTH1	3D-ConvNet + Voting	90.79	90.24	91.42	91.17	91.62	91.04
	3D-ConvNet + LSTM	92.69	96.55	94.25	93.55	94.93	**94.39**
KTH2	3D-ConvNet + Voting	89.14	88.55	89.89	89.45	89.97	89.40
	3D-ConvNet + LSTM	91.50	94.64	90.47	91.31	92.97	**92.17**
	Harris-3D [13] + LSTM	84.87	90.64	88.32	90.12	84.95	87.78

The 3D-ConvNet, combined to majority voting on short sub-sequences, gives comparable results (91.04%) to other deep model based approaches [9,10,26]. We especially note that results with this simple non-sequential approach are almost the same than those obtained in [10], with a 15 times smaller 3D-ConvNet model, and without using neither gradients nor optical flow as input. We also notice that the first step of our model gives relatively stable results on the 5 configurations, compared to the fluctuations generally observed for the other methods [4]. The LSTM contribution is quite important, increasing performances of about 3%. KTH1 improvement ($+3,35\%$) is higher than KTH2, which confirms that LSTM are more suited for long sequences.

In order to point out the benefit of using automatically learned features, we also evaluated the combination of the LSTM classifier with common engineered space-time salient points. This was done by applying the *Harris-3D* [13] detector to each video sequence, and calculating the *HOF* descriptor (as recommended in [27] for KTH) around each detected point. We used the original implementation available on-line[1] and standard parameter settings. A LSTM classifier was then trained taking as input a temporally-ordered succession of descriptors. Obtained results, reported on Table 1, show that our learned 3D-ConvNet features, in addition to their generic nature, perform better on KTH2 than hand-crafted ones, with performances improvement of 4.39%.

To conclude, our two-steps sequence labelling scheme achieves an overall accuracy of 94.39% on KTH1 and 92.17% on KTH2. These results, and others among the best performing of related work on KTH dataset, are reported on Table 2.

Table 2. Obtained results and comparison with state-of-the-art on KTH dataset: methods reported in bold corresponds to deep models approaches, and the others to those using hand-crafted features

Dataset	Evaluation Protocol	Method	Accuracy
KTH1	Cross validation with 5 runs	**Our method**	**94.39**
		Jhuang et al. [9]	**91.70**
		Gao et al. [4]	95.04
		Schindler and Gool [23]	92.70
	Leave-one-out	Gao et al. [4]	96.33
		Chen and Hauptmann [2]	95.83
		Liu and Shah [17]	94.20
		Sun et al. [25]	94.0
		Niebles et al. [19]	81.50
KTH2	Cross validation with 5 runs	**Our method**	**92.17**
		Ji et al. [10]	**90.20**
		Gao et al. [4]	93.57
	Other protocols	**Taylor et al.** [26]	**90.00**
		Kim et al. [12]	95.33
		Ikizler et al. [7]	94.00
		Laptev et al. [13]	91.80
		Dollar et al. [3]	81.20

Table 2 shows that our approach outperforms all related deep model works [9,10,26], both on KTH1 and KTH2. One can notice that our recognition scheme outperforms the HMAX model, proposed by Jhaung et al. [9] although it is of hybrid nature, since low and mid level features are engineered and learned ones are constructed automatically at the very last stage.

[1] Available at `http://www.irisa.fr/vista/Equipe/People/Laptev/download.html`

For each dataset, Table 2 is divided into two groups: the first group consists of the methods which can be directly compared with ours, i.e those using the same evaluation protocol (which is cross validation with 5 randomly selected splits of the dataset into training and test). The second one includes the methods that use different protocols, and therefore those for whom the comparison is only indicative. Among the methods of the first group, to our knowledge, our method obtained the second best accuracy, both on KTH1 and KTH2, the best score being obtained by Gao et al. [4]. Note that the results in [4] corresponds to the average on the 5 best runs over 30 total, and that these classification rates decreases to 90.93% for KTH1 and 88.49% for KTH2 if averaging on the 5 worst ones.

More generally, our method gives comparable results with the best related work on KTH dataset, even with methods relying on engineered features, and those evaluated using protocols which was shown to outstandingly increase performances (e.g leave-one-out). This is a very promising result considering the fact that all the steps of our scheme are based on automatic learning, without the use of any prior knowledge.

5 Conclusion and Discussion

In this paper, we have presented a neural-based deep model to classify sequences of human actions, without a priori modeling, but only relying on automatic learning from training examples. Our two-steps scheme automatically learns spatio-temporal features and uses them to classify the entire sequences. Despite its fully automated nature, experimental results on the KTH dataset show that the proposed model gives competitive results, among the best of related work, both on KTH1 (94.39%) and KTH2 (92.17%).

As future work, we will investigate the possibility of using a single-step model, in which the 3D-ConvNet architecture described in this paper is directly connected to the LSTM sequence classifier. This could considerably reduce computation time, since the complete model is trained once. The main difficulty will be the adaptation of the training algorithm, especially when calculating the retro-propagated error.

Furthermore, even if KTH remains the most widely used dataset for human action recognition, recent works are increasingly interested by other more challenging datasets, which contains complex actions and realistic scenarios. Therefore, we plan to verify the genericity of our approach by testing it on recent challenging datasets, e.g Hollywood-2 dataset [18], UCF sports action dataset [21], YouTube action dataset [16], UT-Interaction dataset [22] or LIRIS human activities dataset[2]. This will allow us to confirm the benefit of the learning-based feature extraction process, since we expect to obtain stable performances on these datasets despite their high diversity, which is not the case of the approaches based on hand-crafted features.

[2] Available at `http://liris.cnrs.fr/voir/activities-dataset/`

References

1. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Action Classification in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6353, pp. 154–159. Springer, Heidelberg (2010)
2. Chen, M.y., Hauptmann, A.: MoSIFT: Recognizing human actions in. surveillance videos. Tech. Rep. CMU-CS-09-161, Carnegie Mellon University (2009)
3. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72 (2005)
4. Gao, Z., Chen, M.-y., Hauptmann, A.G., Cai, A.: Comparing Evaluation Protocols on the KTH Dataset. In: Salah, A.A., Gevers, T., Sebe, N., Vinciarelli, A. (eds.) HBU 2010. LNCS, vol. 6219, pp. 88–100. Springer, Heidelberg (2010)
5. Garcia, C., Delakis, M.: Convolutional face finder: a neural architecture for fast and robust face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(11), 1408–1423 (2004)
6. Gers, F.A., Schraudolph, N.N., Schmidhuber, J.: Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research 3, 115–143 (2003)
7. Ikizler, N., Cinbis, R., Duygulu, P.: Human action recognition with line and flow histograms. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
8. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: International Conference on Computer Vision, pp. 2146–2153 (2009)
9. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: International Conference on Computer Vision, pp. 1–8 (2007)
10. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. In: International Conference on Machine Learning, pp. 495–502 (2010)
11. Kim, H.J., Lee, J., Yang, H.S.: Human Action Recognition Using a Modified Convolutional Neural Network. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 715–723. Springer, Heidelberg (2007)
12. Kim, T.K., Wong, S.F., Cipolla, R.: Tensor canonical correlation analysis for action classification. In: International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
13. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
15. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: IEEE International Symposium on Circuits and Systems, pp. 253–256 (2010)
16. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos in the wild. In: International Conference on Computer Vision and Pattern Recognition, pp. 1996–2003 (2009)
17. Liu, J., Shah, M.: Learning human actions via information maximization. In: International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
18. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: International Conference on Computer Vision and Pattern Recognition, pp. 2929–2936 (2009)

19. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision 79, 299–318 (2008)
20. Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., Barbano, P.E.: Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing 14(9), 1360–1371 (2005)
21. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
22. Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: International Conference on Computer Vision, pp. 1593–1600 (2009)
23. Schindler, K., van Gool, L.: Action snippets: How many frames does human action recognition require? In: International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
24. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: International Conference on Pattern Recognition, vol. 3, pp. 32–36 (2004)
25. Sun, X., Chen, M., Hauptmann, A.: Action recognition via local descriptors and holistic features. In: International Conference on Computer Vision and Pattern Recognition Workshops, pp. 58–65 (2009)
26. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional Learning of Spatio-temporal Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010)
27. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference (2009)