Spatiotemporal Features Learning with 3DPyraNet

Chapter · October 2016

DOI: 10.1007/978-3-319-48680-2_56

2 authors:

Ihsan Ullah
Parthenope University of Naples
20 PUBLICATIONS **39** CITATIONS

SEE PROFILE

Alfredo Petrosino
Parthenope University of Naples
200 PUBLICATIONS **1,171** CITATIONS

SEE PROFILE

Spatiotemporal Features Learning with 3DPyraNet

Ihsan Ullah[1,2](✉) and Alfredo Petrosino[1]

[1] CVPR Lab, University of Napoli Parthenope, Napoli, Italy
{ihsan.ullah,alfredo.petrosino}@uniparthenope.it
[2] Department of Computer Science, University of Milan, Milan, Italy

Abstract. A discriminative approach based on the 3DPyraNet model for spatiotemporal feature learning is proposed. In combination with a linear SVM classifier, our model outperform state-of-the-art methods on two datasets (KTH, Weizmann). Whereas, shows comparable result with current best methods on third dataset (YUPENN). The features are compact, achieving 94.08 %, 99.13 %, and 94.67 % accuracy on KTH, Weizmann, and YUPENN, respectively. The proposed model appears more suitable for spatiotemporal feature learning compared to traditional feature learning techniques; also, the number of parameters is far less than other 3DConvNets.

Keywords: Action recognition · Dynamic scene understanding · Pyramidal neural network · Deep learning

1 Introduction

In real-world, with the passage of time, people in a video and their surrounding change dramatically, resulting in varying pose, occlusion, illumination in each frame, or subject interactions with some object/subject in the surrounding. Local space-time features have been shown useful for recognition tasks such as object and scene recognition (*SR*) [1–8]. These aforementioned techniques have captured peculiar shape and motion in video and provide a representation of events that is relatively independent from their spatiotemporal shifts and scales. For instance, in case of action recognition *(AR)*, 2D hand-crafted features concerning gradient information, optical flow, and brightness information are substituted in video from spatiotemporal extensions of image descriptors, such as 3D-SIFT [9], HOG3D [10], extended SURF [11], or Local Trinary Patterns [12]. These extracted features are anticipated to translate information which is useful for recognition of an action in a numerical form, namely a vector. Additionally, these vectors are trained to form a representation, such as a histogram of most frequent motions of a video, which captures actions that occur at a specific time in a video clip. Definitely, a general representation is learned using the derived representation of a set of labeled training videos. Subsequently, a classifier models a test sample to its closest action class. Therefore, despite the huge variety

J. Blanc-Talon et al. (Eds.): ACIVS 2016, LNCS 10016, pp. 638–647, 2016.
DOI: 10.1007/978-3-319-48680-2_56

of descriptors, it is still intuitive to handle them in a special manner. Classifiers will result in achieving optimal results on specific datasets.

However, video frames contain complex, redundant, and highly variable information. Thus, it is necessary to discover useful features from raw data. Traditional hand-crafted features often require expensive human labor and expert knowledge. Normally they do not generalize well. This motivates the design of an efficient general 'feature learning' technique that can work on different application and datasets. Recently, deep models such as Convolutional-RBM [13], ST-DBN [14], 3DConvNet [15], 3DPyraNet [16], and C3D [8] have been explored to learn and extract spatiotemporal features. These approaches, automatically learn low-to-high level discriminative representations directly from raw videos. Further, feature extraction process is motivated by the fact that classifier require input that is mathematically and computationally convenient to process.

An important aspect of convolutional *DL* models [7,8,13,14,17,18] is the weight learning and sharing concept. Sharing reduces large number of parameters as opposed to conventional *NN* models. Learning parameters in the convolutional models mean learning a kernel shape which is not specific to any neuron as it is slided and shared over the whole image. This reduces the number of parameters, but increases the chance to put burden on those parameters while considering huge amounts of data from videos [16,19]. In addition, most of these models did not follow a biological pyramidal structure, i.e. in these models, while resolution is reduced, kernels and maps actually increase together with the number of layers which violates the biological pyramidal structure, which might be helpful in enhancing the model interpret-ability.

Recently, other models like [20,21] were studied with the common aim to model artificial neural networks for recognition capable to catch the typical biological structure of cortical neural networks. For instance, Phung et al. [21] proposed a pyramidal model *(PyraNet)* where each neuron has a unique kernel obtained from a weight matrix. This weight matrix has the same size as the image/feature map at lower layer. The model is quite similar to *CNN*, however, the difference is in the weight sharing concept and processing i.e. it performs a correlation operation *(Corr)* instead of a convolution to get an output neuron.

This idea was extended to the three dimensional domain in [16] i.e. *3DPyraNet*. It extracts features from both spatial and temporal domain while maintaining pyramid structure for refinement. Thereby it is capable to capture the motion information encoded in multiple adjacent frames. Analyzing traditional *CV* and deep models, we are motivated to propose a generalized model that consists of a new descriptor that replaces local hand-crafted descriptors and histogram generation steps by deep learned representation. Here, we use the *3DPyraNet* model to learn features from raw data and classify them with a linear SVM for *AR* and scene understanding *(SU)*.

The paper is further organized as follows. Section 2 will start with motivational background followed by description of our proposed models for feature extraction. In Sect. 3 we discuss results as well as additional benefits of the proposed model. In the end, Sect. 4 will conclude the paper.

2 Feature Extraction with 3DPyraNet

A short review of *3DPyraNet* will be given in the following section. Subsequently, we will explain the proposed models in coming Subsects. 2.2 and 2.3.

2.1 3DPyraNet

3DPyraNet [16] is inspired from *3DCNN* and *PyraNet*. The key factors that make it different from *3DCNN* is its pyramid structure and the adaptation of weighting scheme that results in correlation. These characteristics are adopted and extended from *PyraNet*. *3DPyraNet* model consist of 5 main layers i.e. (1) input layer, (2) followed by 3-D Correlation (*3D-WeightedSum* or *3DCORR*) layer to learn similar features among input frames/features maps, (3) a 3-D Pooling (*3DPOOL*) layer to reduce resolution for faster processing and to obtain a translation and scale invariant features, (4) another *3DCORR* layer, (5) and finally a fully connected (*FC*) layer for classification as shown in Fig. 1 (without

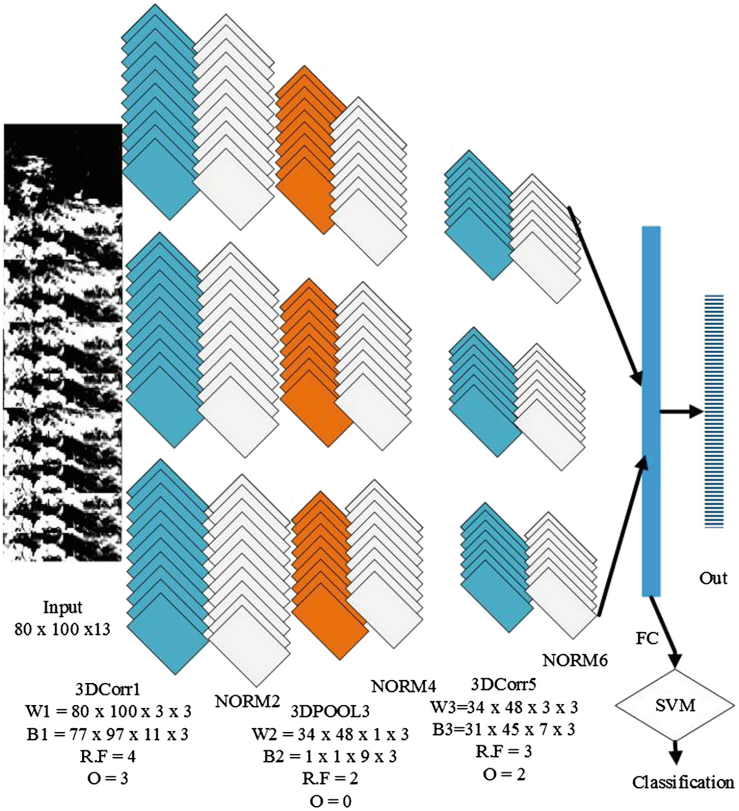

Fig. 1. Proposed model of 3DPyraNet-F. Blue represents Correlation layers, gray represents normalization, and brown represents pooling (Color figure online)

SVM step). Leaky rectified linear unit (*LReLu*) and Sigmoid (*Sig*) are used as activation functions in pyramidal and fully connected layers, respectively. Normalization (zero mean unit variance) is done after every layer to achieve faster convergence and better performance. No sophisticated pre-processing is done as compared to existing *DL* models [17]. However, the given input in case of *AR* was in silhouettes/binary form extracted using [22].

3DPyraNet uses a weight matrix of equal size to the input image/feature map. They have a fixed unique kernel $RF \times RF \times D$ for each output neuron. These kernels have low sharing i.e. in the worst case 1 otherwise depends on overlap (O) and receptive field (RF). Overlap *'O'* represents the number of columns or rows reused in a new adjacent *RF*. Whereas,*RF* can also be considered as the size of the kernel. *3DPyraNet* uses depth of size '$D = 3$' at each step to incorporate the spatial as well as temporal information from the given input frames. To extract more variant and discriminative features, multiple sets of 3-D weight matrices are used. Even in this case, the number of parameters are less than other deep models. This model generates sparse features as compared to convolutional kernel. The training process is performed using same backpropagation with stochastic mini-batch gradient descent approach [16].

A variety of deep architectures can be designed from *3DPyraNet* based on its application area and performance can be enhanced using different input image size, complexity of model, or combination of multiple models. However, we discuss and elaborate mainly two types of models to advere feature extraction, i.e. Late fusion (local) and early fusion (global) an inspiration from work done in [23].

2.2 3DPyraNet-F

Selection of optimal architecture for problem is challenging, since it depends on the specific application. A generalized model was shown in Fig. 1 due to limited space. Mainly, it consists of two *3DCORR* layers, a *3DPOOL*, a *FC* layer, and a linear-SVM classifier layer. Once the convergence is advered, learned features from the last *Norm4* layer are extracted and fused in a single column feature vector. This global/early fusion model is a balanced mix between the spatial and temporal information. These are incorporated in such a way that global information in both spatial and temporal dimensions are progressively accessed by *SVM*. Finally, the trained *SVM* model is used to classify the feature vectors extracted using *3DPyraNet*. One-vs-all criteria is used for classification.

The depth and width of a network and the resulting size of the feature vector depends on the input size, receptive field (RF) and overlap (O) parameters in each layer. (RF) size and (O) are the two main tune-able parameters for handling the performance. We used (RF) size of 4, 2, and 3 with 3, 0, 2 for (O) in layer 1, 2, and 3 of models for *AR*, respectively. Whereas, model for *SR* uses (RF) size of 4, 2, and 4 with (O) as 3, 0, 3 in layer 1, 2, and 3, respectively.

2.3 3DPyraNet-F_M

The difference between *3DPyraNet-F* and *3DPyraNet-F_M* is in the construction of feature vectors. After *3DPyraNet* converges, rather than just concatenating all the feature in one column vector, this model first converts feature maps of same set in one single column vector. Then, the resultant feature vectors are summed together and divided by the number of weight sets to derive their mean vector. This results in a smaller feature vector compared to previous model resulting in faster processing. These features behave as local due to local addition with other features maps. The rest of the model and network architecture is similar to *3DPyraNet-F*.

Third variation of *3DPyraNet-F* can be the size of the network. This difference exists due to different size input image of *AR* and *SR* datasets. Network structures for all models are given in Table 1.

Table 1. Network Structure used for Action (Weizmann(10) and KTH(6)) and Scene (YUPENN(14)) datasets. Feature map size at main Layers

Model	*3DCORR1*	*3DPOOL3*	*3DCORR5*	*FC*	Output
3DPyraNet-F	61 × 45 × 11 × 3	30 × 22 × 9 × 3	27 × 19 × 7 × 3	10773	6/10
3DPyraNet-F_M	61 × 45 × 11 × 3	30 × 22 × 9 × 3	27 × 19 × 7 × 3	3591	6/10
3DPyraNet-F	77 × 97 × 11 × 3	38 × 48 × 9 × 3	35 × 45 × 7 × 3	33075	14

3 Experiments

3DPyraNet-F has been evaluated for recognition tasks. Firstly, we analyze action recognition on the *Weizmann* and *KTH* datasets [4,24]. We will show *3DPyraNet-F* and *3DPyraNet-F_M* enhancement over plain *3DPyraNet* for *AR* datasets. Further, we will compare our model with state-of-the-art hand-crafted feature descriptors as well as feature learners. In second experiment, we examined our models for *SR* on *YUPENN* dataset. In the end, beside its accuracy, another key advantage of our proposed model will be discussed i.e. it's fewer trainable parameters.

Training: Each model is trained on its respective dataset. Table 1 shows feature map size in the form of $w \times h \times m \times s$. Where 'w', 'h', 'm' and 's' represents width, height, number of maps, and weight sets, respectively. *KTH* and *Weizmann* have similar input size i.e. $64 \times 48 \times 13$. Whereas, *YUPENN* has $80 \times 100 \times 13$ with an overlap of 7 images for each clip. Training is done by SGD with mini batch size of 100 clips. We start with a small learning rate i.e. 0.000015 and than decrease it after every 20 epochs by multiplying it with 0.9. The training stops when the testing accuracy stops improving. The *RF* and *O* size are shown in Fig. 1 as well mentioned in Sect. 2.2. A linear-*SVM* is used to analyze discriminative power of the learned features for recognition of action or scene. The *SVM* is trained with Sequential Minimal Optimization (SMO) method of Matlab-2014b Statistical and Machine Learning toolbox [25].

3.1 Action Recognition

KTH and *Weizmann* datasets are relatively small in number, allowing for a more in-depth study. However, they are still challenging to train a deep model as we have to deal with a small training set. *KTH*, an *AR* dataset include 6 classes i.e. Walking, Running, Jogging, Hand clapping, Hand waving, and Boxing. It is a challenging dataset due to outdoor environment and camera movement. We used the same setting and protocol as used in [16]. 3DSOBS [22] is used to extract person from the clip. But due to camera movement very few consecutive clips can be collected, especially in the running case. Therefore, reasonable subsets of random clips of size $64 \times 48 \times 13$ are considered. Features from the *Norm4* layer are extracted and fed to the linear-*SVM*. It classifies each class similar to [4,8,28]. However, we did two types of extraction, i.e. local and global fusion of features as in [23].

In the first case *3DPyraNet-F_M*, vectors consist of 3591 features. Whereas, in the second case *3DPyraNet-F* feature vectors are longer (10733). We achieved mean accuracy of 93.42 % from the binary classifications of one vs all scenarios. Further, (*3DPyraNet-F_M*) enhances the overall performance by 0.67 %. Similarly to [4,17,28], despite fewer training examples, the global fusion (*3DPyraNet-F*) achieved optimal accuracy. In comparison to hand crafted features, our learned feature classified with *SVM* gets better results than *3DHOG*, *Cuboids*, and $Gabor3D + HOG3D$, whereas, almost equal performance are achieved when compared to combination of *HOG*, *HOF*, *MBH*, and *Trajectories* descriptors [3], highlighting more discriminative power of our learned features.

Weizmann is a 10 class *AR* dataset that includes walking, running, jumping, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, and skip. We considered full *Weizmann* dataset and pre-processed it similarly to *KTH*. The same *3DPyraNet-F* and *3DPyraNet-F_M* models were applied. In this case, despite more classes, optimal results were achieved compared to state-of-the-art as shown in Table 2. *3DPyraNet-F* enhances previous results by 8.09 % whereas, *3DPyraNet-F_M* enhanced it further with additional 0.14 %. Only in the case of combination of $HOG + HOF + MBH + Trajectories$, *3DPyraNet-F_M* have a lower accuracy of 0.87 %.

3.2 Scene Understanding

YUPENN is a dynamic scene recognition *(SR)* benchmark. It consists of 420 videos of 14 scene categories i.e. beach, city street, elevator, forest fire, fountain, highway, lighting storm, ocean, railway, rushing river, sky-clouds, snowing, waterfall, and windmill farm. In *SR*, a model has to learn the whole mask rather than a specific portion of the image. As discussed in Sect. 2, *3DPyraNet-F* weight matrix is of equal size of the input image/feature map. Therefore, it could be

Table 2. Accuracies for Action (Weizmann and KTH) and Scene (YUPENN) datasets, Layers represents main layers

Model(classifier)	Weizmann	KTH	YUPENN	Layers
3D-ConvNet [15]	88.26	89.40	-	7
3DCNN [17]	-	90.2	-	6
3DHOG [10]	84.3	91.4	-	-
Cuboids [2]	-	90	-	-
Gabor3D + HOG3D (SVM) [26]	-	93.5	-	-
3DSIFT (SVM) [9]	82.6	-	-	-
HOG + HOF + MBH + Trajectories (SVM) [3]	-	94.2	-	-
C3D (SVM) [8]	-	-	98.1	15
ImageNet [8]	-	-	96.7	8
ST-DBN [27]	-	85.2	-	4
Schuldt (SVM) [4]	-	71.7	-	-
Dollar (SVM) [28]	-	81.2	-	-
3DHOG + Local weighted SVM [29]	100	92.4	-	-
3DPyraNet	90.9	72	-	4
3DPyraNet-F	98.99	93.42	**94.67**	4
3DPyraNet-F_M	**99.13**	**94.083**	-	4
Christoph's [30]	-	-	86.0	-
Theriault's (SVM) [31]	-	-	85.0	-

an ideal case for scene recognition in videos which can also be used as a hint in other recognition tasks.

Model used in this dataset has bigger input size compared to the previous one, i.e. 80 × 100 × 13 hence, resulting in large feature vector of size 33075. We considered an overlap of 7, considering a small number of frames compared to previous models [5,8,32] - for instance, [8] uses 128 × 171 × 16 frames in a clip from which 112 × 112 × 16 random crops were extracted for data augmentation purpose.

Our model achieved best accuracy of 96.2134 % after 25 epochs. However, it achieves mean accuracy of 94.67 % for one-vs-rest classification. Although, it is better than [5,30,31] at huge margin, it does not achieve state-of-the-art performance by 3.43 % fewer accuracy. This could be caused by several reasons; possibly, one resides in the fact that *C3D* [8] is trained on Sports 1-Million videos dataset [23], whereas we trained on the same smaller dataset. In addition, they have high resolution and use augmentation. Although our model is less competitive compared to *C3D* and Imagenet, *3DPyraNet-F* still achieves comparable results, i.e. 94.67 %; a good starting point for future work to test *3DPyraNet-F* on very large scale datasets. Christoph's et al. [32] shows better accuracy than ours by 1.5 % i.e. 96.2 %. However, as opposed to our model where we classify each clip individually, their model uses several complex feature encoding such as FV, LLC, and dynamic pooling to achieve that result on majority voting for video classification.

3.3 Parameters Reduction

After strong success of ImageNet [8,33], models became deeper and deeper. Beside accuracy, their trainable parameters also increased. A separate consideration should be made about the reduction of parameters. The number of parameters is unarguably a substantial issue in application space and the memory cost increases due to the large size of trained models on the disk [8,33–35].

Network in Network (*NIN*) [35] highlighted the issue of reducing parameters, but they achieved it at greater computation cost. As most of the parameters are in fully connected (*FC*) layers, Szegedy et al. [36] uses sparsity reduction complex methodologies for refining those trained models. Han et al. [34] tries to learn connections in each layer instead of weights and then the network is trained again to reduce the number of parameters. *C3D* has about 17.5 M parameters, whereas our model have less than a million parameters (specifically 0.83 M parameters in case for *YUPENN* dataset). Disk occupancy is almost negligible compared to the model trained by C3D; this is of help in embedded systems and mobile devices where the memory usage is a problem.

4 Conclusion

We address the difficulty in learning spatiotemporal features from videos, proposing to adopt the *3DPyraNet* model for feature learning. We show that despite less deeper than state-of-the-art models, our fusion models are capable of learning powerful features by refining the sparse features provided by *3DPyraNet*, achieving competitive results with respect to current best methods on several video analysis benchmarks.

References

1. Laptev, I., Marszaek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
2. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatiotemporal features for action recognition. In: BMVC 2009 - British Machine Vision Conference, pp. 124.1–124.11 (2009)
3. Wang, H., Kläser, A., Schmid, C., Cheng-Lin, L.: Action recognition by dense trajectories. In: CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition, Colorado Springs, United States, pp. 3169–3176. IEEE, June 2011
4. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings - International Conference on Pattern Recognition, vol. 3, pp. 32–36 (2004)

5. Derpanis, K.G., Lecce, M., Daniilidis, K., Wildes, R.P.: Dynamic scene understanding: the role of orientation features in space and time in scene classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1306–1313 (2012)
6. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402 (2012)
7. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatiotemporal features for action recognition with independent subspace analysis. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3361–3368 (2011)
8. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV, pp. 1725–1732. IEEE, June 2015
9. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the ACM International Conference on Multimedia (MM 2007), pp. 357–360 (2007)
10. Klaser, A., Marszalek, M., Schmid, C.: A spatiotemporal descriptor based on 3D-gradients. In: Proceedings of the British Machine Conference, pp. 99.1–99.10 (2008)
11. Willems, G., Tuytelaars, T., Gool, L.: An efficient dense and scale-invariant spatiotemporal interest point detector. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 650–663. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88688-4_48
12. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: IEEE 12th International Conference on Computer Vision, pp. 492–497, September 2009
13. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatiotemporal features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15567-3_11
14. Freitas, N.D.: Deep learning of invariant spatiotemporal features from video. In: Workshop on Deep Learning and Unsupervised Feature Learning in NIPS, pp. 1–9 (2010)
15. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: Salah, A.A., Lepri, B. (eds.) HBU 2011. LNCS, vol. 7065, pp. 29–39. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25446-8_4
16. Ullah, I., Petrosino, A.: A strict pyramidal deep neural network for action recognition. In: Murino, V., Puppo, E. (eds.) ICIAP 2015. LNCS, vol. 9279, pp. 236–245. Springer, Heidelberg (2015)
17. Ji, S., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
18. Simonyan, K., Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos. arXiv preprint arXiv:1406.2199, pp. 1–11, June 2014
19. Uetz, R., Behnke, S.: Locally-connected hierarchical neural networks for gpu-accelerated object recognition. In: NIPS: Workshop on Large-Scale Machine Learning: Parallelism and Massive Datasets, Whistler, Canada, pp. 10–13, December 2009
20. Cantoni, V., Petrosino, A.: Neural recognition in a pyramidal structure. IEEE Trans. Neural Netw. **13**(2), 472–480 (2002)
21. Phung, S.L., Bouzerdoum, A.: A pyramidal neural network for visual pattern recognition. IEEE Trans. Neural Netw. Publ. IEEE Neural Netw. Counc. **18**(2), 329–343 (2007)

22. Maddalena, L., Petrosino, A.: The 3dsobs+ algorithm for moving object detection. Comput. Vis. Image Underst. **122**, 65–73 (2014)
23. Karpathy, A., Leung, T.: Large-scale video classification with convolutional neural networks. In: Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
24. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision (ICCV 2005), vol. 1, pp. 1395–1402 (2005). Vol. 2
25. MATLAB: Matlab version 8.4.0.150421 (R2014b). The MathWorks Inc., Natick, Massachusetts (2014)
26. Maninis, K., Koutras, P., Maragos, P.: Advances on action recognition in videos using an interest point detector based on multiband spatiotemporal energies. In: 2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27–30, 2014, pp. 1490–1494 (2014)
27. Chen, B., Ting, J.A., Marlin, B., de Freitas, N.: Deep learning of invariant spatiotemporal features from video. In: NIPS 2010 Deep Learning and Unsupervised Feature Learning Workshop (2010)
28. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatiotemporal features. In: Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS 2005, pp. 65–72 (2005)
29. Weinland, D., Özuysal, M., Fua, P.: Making action recognition robust to occlusions and viewpoint changes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 635–648. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_46
30. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spacetime forests with complementary features for dynamic scene recognition. In: BMVC (2013)
31. Theriault, C., Thome, N., Cord, M.: Dynamic scene classification: Learning motion descriptors with slow features analysis. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2603–2610, June 2013
32. Feichtenhofer, C., Pinz, A., Wildes, R.: Bags of spacetime energies for dynamic scene recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2681–2688 (2014)
33. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
34. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. CoRR abs/1506.02626 (2015)
35. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR abs/1312.4400 (2013)
36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR, USA, June 7–12, pp. 1–9 (2015)