A Strict Pyramidal Deep Neural Network for Action Recognition

Ihsan Ullah and Alfredo Petrosino(✉)

Department of Computer Science, University of Milan, Milan, Italy
{ihsan.ullah,alfredo.petrosino}@uniparthenope.it

Abstract. A human action recognition method is reported in which pose representation is based on the contour points of the human silhouette and actions are learned by a strict 3d pyramidal neural network ($3DPyraNet$) model which is based on convolutional neural networks and the image pyramids concept. $3DPyraNet$ extracts features from both spatial and temporal dimensions by keeping biological structure, thereby it is capable to capture the motion information encoded in multiple adjacent frames. One outlined advantage of $3DPyraNet$ is that it maintains spatial topology of the input image and presents a simple connection scheme with lower computational and memory costs compared to other neural networks. Encouraging results are reported for recognizing human actions in real-world environments.

1 Introduction

Despite advances in image recognition, action recognition is still challenging as it contains insufficient information for proper classification of an action. Some of the well-known models [1–9] achieved 90+% accuracy on different datasets under their respective targeted scenarios. In real-world scenarios, in most of the cases, human and their surrounding changes dramatically, resulting in angle change, occlusions and interactions and performance of such approaches drop when the dataset or scenario changes.

Recent neural network based methods have been developed by going deeper for learning more discriminative and different features [10]: the functions that can be represented by a k-depth architecture might require an exponential number of computational elements to be represented by a $(k-1)$-depth architecture. This is mainly why in the last years algorithms based on deep learning approaches achieved resounding success in the community of computer vision and in the field of action recognition [11–15].

Specifically, 3D convolutional neural networks have been used to learn spatio-temporal features directly from the raw images [16,17]. They increase kernels and maps as the number of layer increases. An important aspect of convolutional deep models is the weight learning and sharing concept that reduces large number of parameters compared to conventional fully connected neural network models. Learning parameters in the convolutional models are in a kernel shape which are not specific to any neuron; rather, they are slided and shared over the

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 236–245, 2015.
DOI: 10.1007/978-3-319-23231-7_22

whole image and network. It reduces the number of parameters, but increases the chance to put burden on those parameters while considering huge amounts of data from videos. Considering that the number of computational elements strongly depends on the number of training examples available for tuning the network, an insufficiently deep architecture might bring to poor generalization capabilities. So, the main limitation lies in the fact that entire images need to be used for training the network, so determining a very high computational cost which makes often unfeasible the usage of such approaches in real applications.

Our work is inspired by the idea that early models strictly pyramidal and following strict biological structure [18–20] may turn to be a possible solution. In this paper, we will be focusing on two aspects: one proposing a new model that learns features from input till output without any handcrafted features, and secondly proposing and re-utilizing such a weighting scheme that can work better and learn discriminative features for recognizing human actions in videos. In coming section we will discuss a strict pyramidal neural network known as $PyraNet$ that we will utilize for designing our strict pyramidal architecture for action recognition.

The contribution of this paper is twofold. First, introducing 3D strictly pyramidal neural network ($3DPyraNet$) model that will not violate biological neural network structure and which will use a new weighting scheme that extracts discriminative spatial and temporal information. Second, the model is tailored for action recognition, demonstrating, although hard constraints are imposed on the model, it is still able to properly recognize actions.

The paper is further organized as follows. Section 2 will give slight motivational background of why we proposed such a model. Further details about existing techniques that are modified, combined and enhanced in our proposed model are given in sub-sections. Section 3 will show and discuss results achieved from proposed models. Section 4 will conclude the paper.

2 3DPyraNet

We adopted a strict 3D pyramidal architecture based on decision making pyramidal structure of a brain through feature maps. T Furthermore, to capture actions as a whole in videos, we adopted a similar weighting scheme as used in $PyraNet$ that will be discussed in section 2.2. These parameters will be learned from input till output using a modified structure of traditional backpropagation algorithm. To take advantage of temporal information in videos, we adopted 3D structure by taking inspiration from 3D Convolutional Neural Network ($3DCNN$) model [14]. The aim is to show that a strict pyramidal structure can enhance performance compared to unrestricted models even with simple structure, fewer feature maps and hidden layers. In coming sub-section we will first explain existing $Pyranet$ model and than the weighting scheme before going to our proposed 3D architecture.

2.1 PyraNet

The *PyraNet* model [19] was inspired by the pyramidal neural network model reported in [18] with 2D and 1D layers. The diversity from the original model presented in [18] was that the coefficients of a receptive field were adaptive and it performed feature extraction and reduction in lower 2D layers. These were followed by a 1D layer at the top for classification of an image.

This model is also similar to CNN if we remove pooling layers from a CNN [15,21]. Specifically: 1) the model is connected directly to pixels in the input image (no preprocessing holds); 2) neurons are connected only to local regions (locally connected network); and 3) layers form a reduced representation of the preceding layers (deep is the rule of thumb).

However there were two main differences. Firstly it does not perform convolution rather weighted sum (WS) operation or correlation over the receptive field. Secondly, weights are not in the form of a kernel that slides over the whole image; rather each output neuron has a local unique kernel specifically assigned to it. These kernels are based on input neurons in a receptive field and their corresponding weights in a weight matrix. This results in a unique locally connected kernel for each output neuron. This will be further discussed in next section. Further, in *PyraNet* the dimensions were reduced by the stride of the kernel at each layer.

2.2 Weighting Scheme

An important part in popular convolutional deep models is their weight-sharing concept that gives an edge over other neural network models. This property reduces large amount of learning parameters; however, it increases burden on those fewer parameters. We adopted the same weighting scheme as used in *PyraNet*. In *PyraNet* each neuron has its own unique weight. These are learned using backpropagation technique and stochastic mini-batch gradient descent approach. This unique weight scheme results in weight matrix that has same size as input image or feature map at a lower layer. At the time of computation, each output neuron gets a unique kernel based on calculated receptive field.

This approach is modified for 3D structure by using more weight matrices at a time to incorporate the temporal information from the given input frames. In order to capture further different type of features, several sets of weight matrices WS are used. We randomly initialize these weights at each layer taking care of suggested techniques stated in literature for corresponding activation functions used at those layers. Initially, feature maps produced by WS kernels were sparse as compared to convolutional kernel. But later by training the model, they become similar to smooth blurred images of the input sequences. This 3D weight matrix approach for weight-sharing is different than traditional one. The weight sharing is very minimal in this approach, i.e. in worst case $1/n$ where n is the size of the receptive field kernel. Each neuron has its weight parameter that is locally shared whenever that neuron is used in a receptive field of an output neuron.

2.3 Proposed Architecture

The basic $3DPyraNet$ model has three hidden layers in combination with pooling layer as shown in Figure 1. Unlike deep models for videos [14], no specific or sophisticated pre-processing is done and we adopt a silhoutte-based approach. In general, the temporal part gives correlation between the objects or actions in consecutive frames of a video. Therefore, the first hidden layer is a 3D weighted sum WS or correlation layer ($L1WS$) shown in Figure 1. It results in maps containing spatial as well as temporal information extracted from the given sequence of input frames at input layer. This WS layer is a pure correlation operation among the given neurons and weights in a receptive field of a frame and weight matrix as shown in equation 3.

$$y^l_{u,v,z} = f_l(\sum_{i \in R_{u,v,z}} \sum_{j \in R_{u,v,z}} \sum_{m \in R_{u,v,z}} (W^l_{i,j,m} \circ x^{l-1}_{i,j,m}) + b^l_{u,v,z}) \tag{1}$$

where f_l represents the activation function used at layer l, (u, v) the neuron location at z output map generated by the set of input maps m from layer $l-1$. To compute the receptive field of neuron (u, v) i.e. $R^{l,m}_{(u,v,z)}$, we use equation 2. As in our case, the input map x at layer $l-1$ and the weight matrix W for layer l have the same size, (i, j) represents the same corresponding location in matrix x and W. Equation 2 determines the local receptive field for neurons and their respective weight parameters. Concerning biases, differently from $CNNs$, we do not use one bias for each output feature map, but one bias for each output neuron in the output feature map.

$$R^{l,m}_{u,v,z} = \begin{cases} (i,j,z) \mid (u-1) \times g_l + 1 \leq i \leq (u-1) \times g_l + r_l; \\ (v-1) \times g_l + 1 \leq j \leq (v-1) \times g_l + r_l; \\ z \leq m \leq m + r_l - g_l \end{cases} \tag{2}$$

The reason behind using a correlation $\circ$ operation in equation 3 rather than a convolution is that correlation extracts and collect similarity. Since action recognition is defined as recognition of consecutive, almost similar activity or pose of a human body over a continuous time span, correlation or weighted sum operation is most suitable for recognizing similar actions in videos due to the correlation existence in consecutive frames. The WS layer has two main tunable parameters, i.e. receptive field size and stride for handling the performance. We used three sets of $3D$ weight matrix in order to extract different types of features from the actual input. The set of weights remains the same whereas their size reduces throughout the network until $1D$ layer. In addition, maps decrease by two in each set as we go deeper in the network. At each layer, after passing the feature maps through activation function, the output maps are whitened in order to allow fast convergence by regulating their saturation. To capture global as well as local discriminative features among consecutive correlated feature maps, $L1WS$ is followed by a $3D$ temporal pooling layer ($L2P$) which not only reduces spatial resolution but also, due to $3D$ pooling associated with the temporal domain, but also leads to more discriminative feature maps.

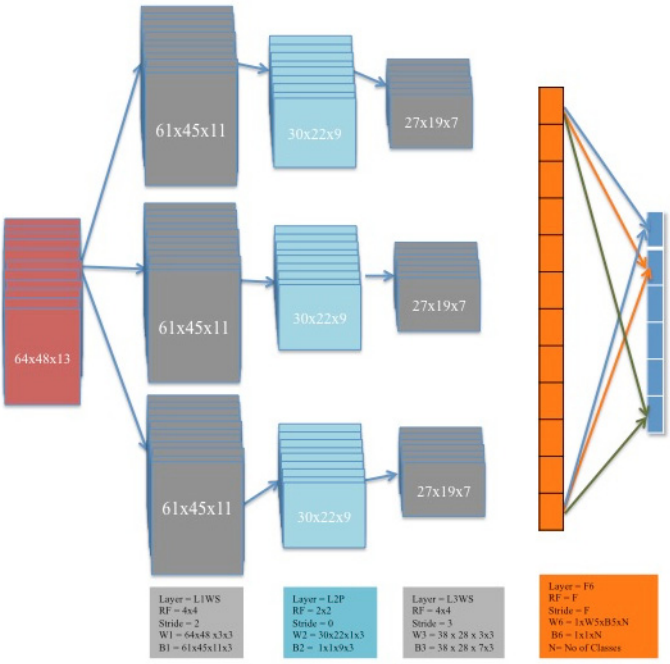


Fig. 1. Proposed model of 3DPyraNet

$$y^l_{u,v,z} = f_l(W^l_{u,v,z} \times \max_{(i,j,m)\in R_{u,v,z}} y^{l-1}_{i,j,m} + b^l_{u,v,z}) \tag{3}$$

The third layer is again a correlation layer ($L3WS$), whose output is converted to a $1D$ column feature vector that is used as a fully connected layer for classification. The overall $3DWS$ and pooling layers extract discriminative features by capturing the motion information encoded in multiple adjacent frames. Weight update is done using conventional back-propagation algorithm, where stochastic gradient descent approach is applied for weights update. We use a small learning value and reduce it by a factor of 10% after 10 epochs. Cross entropy error function is used to reduce the error. In the next section we will discuss our results achieved after experimenting on different state-of-the-art action recognition datasets.

3 Results and Discussion

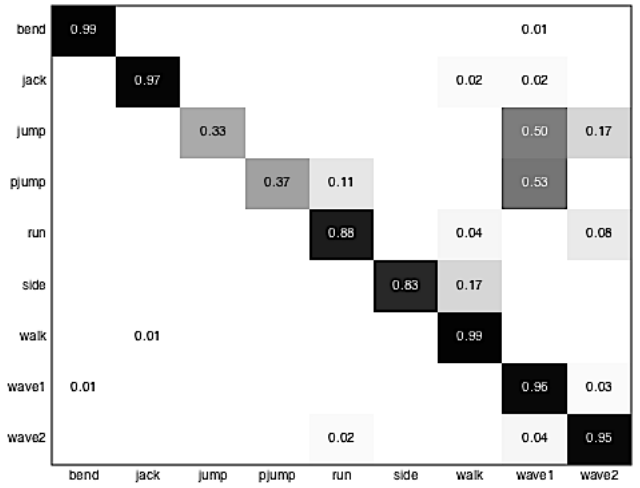
The $3DPyraNet$ has been evaluated on Weizmann and KTH datasets. Weizmann is a good starting dataset for evaluating performance of a network. It is smaller in comparison to others in terms of action sequences. However, it provides ten types of quite similar human actions, i.e. walking, running, jumping,

galloping sideways, bending, one-hand wave, two-hands wave, jump in place, jumping jack, and skip. Each action is done by nine actors in different scenarios that make it a complex task for recognition having only few short videos for each category which is not a good scenario rather challenging for DL models. KTH is another popular big and complex dataset that contains six actions done by 25 actors. It provides 2391 sequences in four different scenarios along with camera movement that results in different resolutions. We used a sequence of 13 consecutive frames of size 64×48 to represent an action for both datasets.

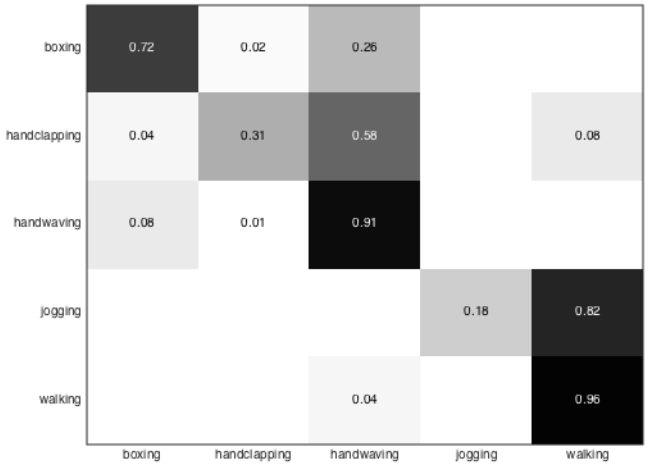
We carried out two types of experiments 1) to check the efficiency of proposed WS layers with simple activation functions and 2) later combining it with pooling and using advance rectification functions. Therefore, first we evaluated the effect of WS layer. We used a network with two WS layers and a fully connected layer to classify amongst ten classes. The output of each WS layer was passed through an activation function, i.e. sigmoid or tangent and then normalized throughout the network learning. Initial learning was not smooth and took around 450 epochs to converge. This provided an accuracy of 80% on the training set and 70% on the testing set.

As in most deep models, pooling plays an important role by providing translation invariance as well as reducing the dimensions. In addition, for faster convergence, avoidance of local minima, and improvement in performance; an extension of rectified linear unit known as leaky rectified linear units ($LReLu$) [22] is utilized. This $LReLu$ in contrast to $ReLu$ allows a small non-zero gradient when the neuron activity is less than or equal to zero. This property overcomes the limitation of $ReLU$ and updates the weights even if stuck within zeros. Therefore in our model, we adopt temporal ($3D$) pooling at ($L2P$) and $LReLu$ in combination to WS layers. This resulted in higher accuracy, i.e. 87% and 80.5% respectively for training and testing along with faster convergence i.e. within 200 epochs. Moreover, learning behaviour during training was quite smooth compared to the previous model. Furthermore, when we used voting scheme for classification of videos based on classified action sequences, the result increased by two to three percent.

We compared $3DPyraNet$ with deeper models having five to eight hidden layers. To better evaluate our model we reported the mean accuracy on 5 splits of training and testing datasets selected from same Weizmann database as adopted for evaluation of several models. Indeed, to cross validate the results, we randomized the data in same proportion keeping in mind that equal number of sequences should exist for the small number of sequences e.g. 'skip' or 'running'. So doing, no overtraining is observed. Obtained results, corresponding to 5 randomly selected training/test configurations are reported on Table 1. We achieve 90.9% accuracy considering all ten classes in the dataset as provided in Table 1 (a). However, videos containing action 'skip' were short, so letting an unbalanced dataset; this is why other approaches reported in literature did not use this category in their experiments. Our model was unable to learn and classify the 'skip' category according to our expectations, owing to lack of training data. Therefore, if we neglect the skip category, accuracy increases to 92.46% as shown in Table 1 (b.)In case of 'pjump', the same problem of having fewer training sequences arose, that



(a)



(b)

Fig. 2. Confusion matrices: (a) Weizmann without 'skip' (b) KTH without 'running'.

resulted in poor performance. However, for the rest of the categories, $3DPyraNet$ shows optimal results in both sequence and video classification shown on left of Figure 2. Results on Weizmann are comparable with the state-of-the-art model $3DCNN$, which is impressive considering fewer number of hidden layers and having no sophisticated pre-processing for extracting hard coded features.

The second dataset adopted in our experiments is KTH. The same criteria that took 9 out of 25 person's videos for testing, as stated in literature, was adopted. We randomly selected a total of 200 sequences from them of size

Table 1. (a) Mean accuracy of five random data setups, (b) Proposed Vs. Others for Weizmann and KTH datasets

(a)

SetUp	Accuracy(%)
1	90.5
2	92
3	90
4	90.5
5	91.3
Mean	90.9

(b)

Model(classes)	Weizmann (%)	KTH(%)	Layers
3DCNN	88.26[16]	90.2 [14]	6
3DPyraNet (all)	90.9	72	4
3DPyraNet (all-1)	92.46	74.23	4
ST-DBN	-	85.2	4
GRBM	-	90.0	-
Schuldt [6]	-	71.7	-
Dollar [23]	-	81.2	-
Alexandros [9](all)	90.32	-	-
Alexandros [9](all-1)	92.77	-	-

$13 \times 64 \times 48$. It should be noted that in our initial experiments we faced the same problem for 'running' class videos, i.e. having fewer frames than minimum requirement of 13 due to fast movement of the person or camera zooming scenarios. We achieved 72% accuracy over six classes. If we remove the 'running' class due to insufficient training data, the accuracy grows to 74.23% (see confusion matrix on right of Figure 2). Table 1 (b) shows the comparison of our proposed model with the state-of-the-art models reported in literature for Weizmann and KTH datasets. In case of Weizmann we overcome reported best result of 88.26% with an average of 91.07% from ten tests using the same dataset and number of consecutive input frames [16]. On the other hand for KTH dataset, $3DPyraNet$ did not show better result as provided by $3DCNN$ [14], but still it shows comparable results to some other complex models. One of the most plausible reason is that deep models need more data to have better understanding of their respective problems. $3DCNN$ [14] used ROI's sequences extracted and classified by another CNN based methodology. Unlike aforementioned, we used only silhoutte based recognition - extracted by the background subtraction model SOBS [24] in our case - and than extracted the ROI containing human. This may contain half, not centered or unaligned ROIs as input. This can greatly affect the learning process and may have high impact in reducing the classification rate compared with $3DCNN$. Despite current performance, there is room for further study. The future work can be done by using the full datasets of KTH, UT interaction dataset, UCF, TRECVID etc. to evaluate the performance of proposed network model with large number of training and testing data.

4 Conclusions

A strict pyramidal 3D neural network has been proposed that gets raw input frames from videos as input and is able to learn features in fewer layers due to its pyramid structure. It provided better results in case of Weizmann and comparable results with KTH datasets. We are verifying the generality of our model

by testing it on recent larger and challenging datasets like UCF sports, Youtube action, and UT-Interaction datasets. This will help in proving benefits of using strictly pyramidal structure instead of non-pyramidal structure for learning a powerful model, since the model is aimed to obtain good performance despite the complexity and diversity of these datasets.

References

1. Schindler, K., Van Gool, L.: Action Snippets: How many frames does human action recognition require? In: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
2. Yang, X., Tian, Y.L.: Action Recognition using super sparse coding vector with spatio-temporal awareness. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part II. LNCS, vol. 8690, pp. 727–741. Springer, Heidelberg (2014)
3. Liu, W., Wang, Z., Tao, D., Yu, J.: Hessian regularized sparse coding for human action recognition. In: He, X., Luo, S., Tao, D., Xu, C., Yang, J., Hasan, M.A. (eds.) MMM 2015, Part II. LNCS, vol. 8936, pp. 502–511. Springer, Heidelberg (2015)
4. Melfi, R., Kondra, S., Petrosino, A.: Human activity modeling by spatio temporal textural appearance. Pattern Recognition Letters **34**(15), 1990–1994 (2013)
5. Efros, A.-A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proceedings Ninth IEEE International Conference on Computer Vision, pp. 726–733. IEEE Computer Society (2003)
6. 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)
7. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Effective codebooks for human action representation and classification in unconstrained videos. IEEE Transactions on Multimedia **14**(4 PART 2), 1234–1245 (2012)
8. Wu, D., Shao, L.: Silhouette analysis-based action recognition via exploiting human poses. IEEE Transactions on Circuits and Systems for Video Technology **23**(2), 236–243 (2013)
9. Chaaraoui, A.A., Climent-Prez, P., Flrez-Revuelta, F.: Silhouette-based human action recognition using sequences of key poses. Pattern Recognition Letters **34**(15), 1799–1807 (2013). Smart Approaches for Human Action Recognition
10. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
11. 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, Part VI. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010)
12. Freitas, N.D.: Deep learning of invariant spatio temporal features from video. In: Workshop on Deep Learning and Unsupervised Feature Learning in NIPS, pp. 1–9 (2010)
13. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal 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)

14. Ji, S., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35**(1), 221–31 (2013)
15. 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)
16. 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)
17. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1), 221–231 (2013)
18. Cantoni, V., Petrosino, A.: Neural recognition in a pyramidal structure. IEEE Transactions on Neural Networks **13**(2), 472–480 (2002)
19. Phung, S.L., Bouzerdoum, A.: A pyramidal neural network for visual pattern recognition. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council **18**(2), 329–343 (2007)
20. Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Networks **1**(2), 119–130 (1988)
21. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
22. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. ICML, vol. 30 (2013)
23. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal 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)
24. Maddalena, L., Petrosino, A.: The 3dsobs+ algorithm for moving object detection. Computer Vision and Image Understanding **122**, 65–73 (2014)