Scene Understanding using Deep Learning

Farzad Husain[a,c], Babette Dellen[b], Carme Torras[a]

[a]*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028, Barcelona, Spain.*
[b]*RheinAhrCampus der Hochschule Koblenz, Joseph-Rovan-Allee 2, 53424 Remagen, Germany.*
[c]*Catchoom, Barcelona, Spain.*

Abstract

Deep learning is a type of machine perception method that attempts to model high-level abstractions in data and encode them into a compact and robust representation. Such representations have found immense usage in applications related to computer vision. In this chapter we introduce two such applications, i.e., semantic segmentation of images and action recognition in videos. These applications are of fundamental importance for human-centered environment perception.

Keywords: Deep learning, semantic segmentation, action recognition.

1. Introduction

Automation based on artificial intelligence becomes necessary when agents such as robots are deployed to perform complex tasks. Detailed representation of a scene makes robots better aware of their surroundings, thereby making it possible to accomplish different tasks in a successful and safe manner. Tasks which involve planning of actions and manipulation of objects require identification and localization of different surfaces in dynamic environments [1, 2, 3].

The usage of structured light based depth sensing devices has gained much attention in the past decade. This is because they are low-cost and capture data in the form of dense depth maps, in addition to color images. Convolutional Neural Networks (CNNs) provide a robust way to extract useful information from the data acquired using these devices [4, 5, 6, 7]. In this chapter we will discuss the basic idea behind standard feed forward CNNs (Section 2) and their application in semantic segmentation (Section 3) and action recognition (Section 4). Further in depth analysis and state-of-the-art solutions for these applications can be found in our recent publications [6] and [7].

2. Convolutional Neural Networks

Convolutional Neural Networks are directed acyclic graphs. Such networks are capable of learning highly non-linear functions. A neuron is the most basic unit inside a CNN. Each layer inside a CNN is composed of several neurons. These neurons are

hooked together so that the output of neurons at layer l becomes the input of neurons at layer $l+1$, i.e.,

$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)}), \tag{1}$$

where $W^{(l)}$ is the weight matrix of layer l, $b^{(l)}$ is the bias term and f is the activation function. The activation for layer l is denoted by $a^{(l)}$. Training a CNN requires learning W and b for each layer such that a cost function is minimized. Formally, given a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ of m training examples, the weights W and bias b need to be determined that will minimize the cost, i.e., the difference between the desired output y and the actual output $f_{W,b}(x)$. The cost function for one training example is defined as:

$$J(W,b;x,y) = \frac{1}{2}||h_{W,b}(x) - y||^2, \tag{2}$$

where, $h(x)$ gives the activations of the last layer.

The minimization is done iteratively using a gradient descent approach which involves the computation of partial derivatives of the cost function with respect to the weights and updating the weights accordingly. One iteration of gradient descent updates the parameters W, b as:

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial}{\partial W^{(l)}} J(W,b), \tag{3}$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial}{\partial b^{(l)}} J(W,b). \tag{4}$$

The backpropagation algorithm is used to compute the partial derivatives of the cost function.

Fully connected layers have all the hidden units connected to all the input units. This increases the number of connections tremendously when dealing with high dimensional data such as images. If we consider the image size as its dimension then connecting each input pixel to each neuron becomes computationally expensive. An image as small as 100×100 pixels would need $10^4 \times N$ connections at the input layer, where N is the number of neurons at the first layer.

Convolutional layers allow to build sparse connections by sharing parameters across neurons. Compared to fully connected layers, convolutional layers have fewer parameters, so they are easier to train. This comes at the cost of a slight decrease in performance [8]. Commonly used CNNs for image recognition consist of several layers of convolution followed by a few fully connected layers [8, 9]. Such networks are often termed *deep networks*.

3. Semantic labeling

Dense semantic labeling of a scene requires assigning a label to each pixel in an image. The label must represent some semantic class. Such labeling is also referred to as object class segmentation because it divides the image into smaller segments, where each segment represents a particular class. Semantic labeling is challenging because

naturally occurring indoor and outdoor scenes are highly unconstrained, leaving little room for discovering patterns and structures. The semantic classes can be abstract such as "furniture" or more descriptive such as "table", "chair" etc. The more descriptive labeling we aim to achieve, the harder it becomes.

Convolutional Neural Networks provide a robust way to learn semantic classes. A CNN architecture used for semantic labeling typically consists of convolution and pooling layers only [6, 5]. The number of channels in the last layer is equal to the number of object classes that we want to learn. Figure 1 shows a basic example of a deep network architecture used for semantic segmentation.

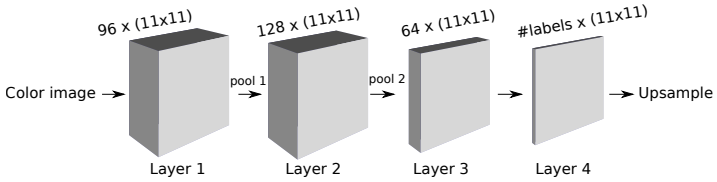

Figure 1: A model architecture for pixelwise semantic labeling. The network consists of 4 convolutional layers and 2 max pooling layers. The output Layer 4 has the number of channels equal to the number of class labels that needs to be learned. The filter sizes in each layer have been set to (11×11). Finally, the output feature maps obtained are upsampled to be of the same size as the input image.

A CNN is usually trained to minimize a multiclass cross entropy loss function [4]. Formally given an image X of a scene, the objective is to obtain a label $\hat{y}_p \in \mathscr{C}$ for each pixel location $\mathrm{x}_p \in X$ that corresponds to the object class at the pixel location. The loss function L can now be written as:

$$L = -\sum_{i \in X} \sum_{b \in \mathscr{C}} c_{i,b} \ln(\hat{c}_{i,b}),$$

where, $\hat{c}_{i,\cdot}$ is the predicted class distribution at location i, and $c_{i,\cdot}$ is the respective ground truth class distribution.

3.1. Related research

Several improvements in the past have been proposed to learn rich features from color images. One approach is to use image region proposals for training CNNs [10]. Another approach is to explore contextual information between different image segments [11]. Classification of superpixels at multiple scales has also been investigated in the past [12]. Another possibility is to train a network end-to-end by attaching a sequence of deconvolution and unpooling layers [13]. Recently, a joint training of a decoupled deep network for segmentation and image classification was shown to facilitate semantic segmentation results [14].

Different ideas for semantic labeling have been proposed which also utilize the depth information in RGB-D images. A depth normalization scheme where the furthest point is assigned a relative depth of one is proposed in [15]. Using height above the ground plane as an additional feature was investigated in [16, 17]. A bounding hull

heuristic to exploit indoor properties was proposed in [15]. In our recent study [6], we proposed a novel feature *distance-from-wall*. This feature was used to highlight objects that are usually found in close proximity to the walls detected in indoor scenes.

Table 1: Individual classes of NYU v2 (four classes) and overall average.

	Accuracy (%)					
Method	floor	struct	furniture	prop	class avg.	pixel avg
Couprie et al. [18]	87.3	86.1	45.3	35.5	64.5	63.5
Khan et al. [19]	87.1	**88.2**	54.7	32.6	69.2	65.6
Stückler et al. [20]	90.7	81.4	68.1	19.8	70.9	67.0
Müller and Behnke [21]	94.9	78.9	71.1	42.7	72.3	71.9
Wolf et al. [22]	**96.8**	77.0	70.8	45.7	72.6	74.1
Eigen and Fergus [4] (AlexNet)	93.9	87.9	**79.7**	55.1	79.1	**80.6**
Husain et al. [6]	95.0	81.9	72.8	**67.2**	**79.2**	78.0

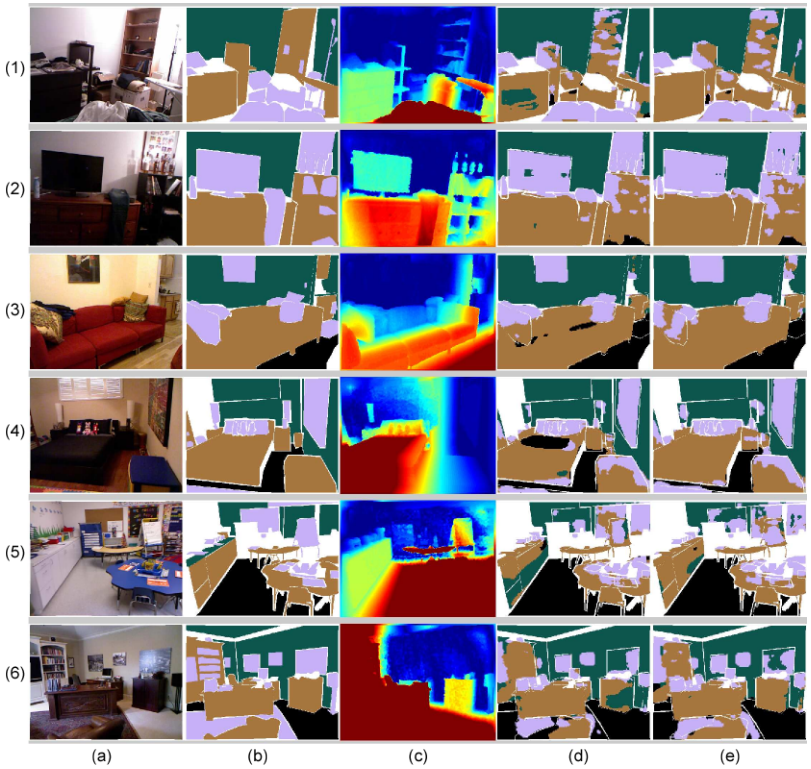

Figure 2: Some examples of semantic labeling, (a) color image, (b) ground truth labeling, (c) distance-from-wall, (d) predicted labels without distance-from-wall and (e) predicted labels with distance-from-wall. White color in Figs. (b), (d) and (e) represents the unknown label. Figure reproduced from Husain et al. [6].

Commonly used datasets for benchmarking different image segmentation approaches include the PASCAL Visual Object Classes dataset [23], and for the RGB-D data include the NYU-v2 dataset [24] and the SUN RGB-D dataset [25]. Table 1 shows some state-of-the-art results for the NYU-v2 dataset for four semantic classes as defined by Silberman and Fergus [24]. These classes are defined according to the physical role they play in the scene, i.e., "floor", "structures" such as walls, ceilings, and columns; "furniture" such as tables, dressers, and counters; and "props" which are easily movable objects. Figure 2 shows some examples of semantic labeling results achieved by Husain et al. [6].

4. Action Recognition

Recognizing human actions from videos is of central importance in understanding dynamic scenes. Recognition is typically performed by processing a video containing a particular action and predicting a label as the output. Action recognition is a challenging task because similar actions can be performed at different speeds, recorded from different viewpoints, lighting conditions and background.

Convolutional Neural Networks provide a way to recognize actions from videos. The most basic approach using CNNs involve treating each frame of the video as an image and predicting the action for each frame followed by averaging over all the predictions. Figure 3 shows a basic action recognition pipeline using a CNN.

It has been shown in the past that a CNN model trained on one dataset can be transferred to other visual recognition tasks [26, 27]. We also see this transfer learning technique being applied successfully for recognizing actions. This is achieved by using a pretrained image recognition model for the individual frames of videos [7, 28, 29].

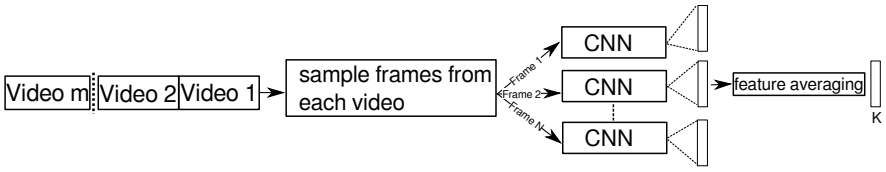

Figure 3: Illustration of a CNN network used for recognizing actions. Features from each frame are extracted using a CNN and averaged. K is the number of action categories. The final feature vector gives a probability for each action.

4.1. Related research

Attempts have been made to make action recognition invariant to different kinds of situations. This includes the usage of optical flow as additional information [28] or using 3D (spatio-temporal) convolutional kernels [7, 30]. Recurrent Neural Networks have also been explored to learn from long term dependencies in different types actions [31]. Learning actions representation in an unsupervised way has also been proposed [32]. This involved using Long Short Term Memory (LSTM) networks for encoding videos and afterwards reconstructing them. Recently, a concept of dynamic

Table 2: Average accuracy on the UCF-101 dataset (3-fold).

Algorithm	Accuracy
CNN with transfer learning [29]	65.4%
LRCN (RGB) [36]	71.1%
Spatial stream ConvNet [28]	72.6%
LSTM composite model [37]	75.8%
C3D (1 net) [30]	82.3%
Temporal stream ConvNet [28]	83.7%
C3D (3 nets) [30]	85.2%
Combined ordered and improved trajectories [38]	85.4%
Stacking classifiers and CRF smoothing [39]	85.7%
Improved dense trajectories [40]	85.9%
Improved dense trajectories with human detection[41]	86.0%
2D followed by 3D convolutions [7]	86.7%
Spatial and temporal stream fusion [28]	88.0%

image was proposed [33]. The dynamic image encodes the temporal evolution of a video and is used for the task of action recognition.

In our recent study, we demonstrated how human action recognition can be achieved using the transfer learning technique coupled with a deep network comprising 3D convolutions [7].

Commonly used datasets for benchmarking different approaches include the UCF-101 dataset [34], the HMDB dataset [35] and the Sports 1M dataset [29]. Table 2 shows some state-of-the-art results for the UCF-101 dataset for 101 action classes. Figure 4, reproduced from one of our previous studies [7], shows the top-5 predictions for selected sequences from the UCF-101 dataset. It can be observed that the actions performed in visually similar environments are often predicted with a high probability. Consider for example, Fig. 4(c6) vs. Fig. 4(b3). Figure 5 shows the confusion matrix for the action sequences from the HMDB dataset using our approach as described in [7]. It can be observed that similar actions such as "sword exercise" and "draw sword" have some degree of confusion.

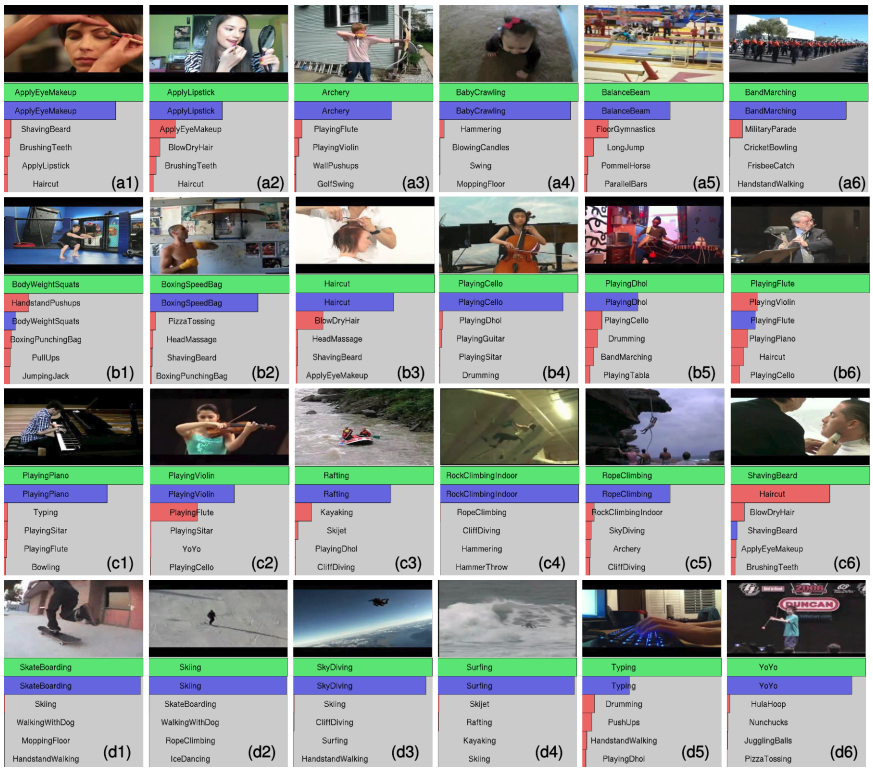

Figure 4: Some results for top-5 predicted action labels for the UCF-101 dataset [34]. First row (green color) shows the ground-truth followed by predictions in decreasing level of confidence. Blue and red show correct and incorrect predictions, respectively. The figure is taken from Husain et al. [7].

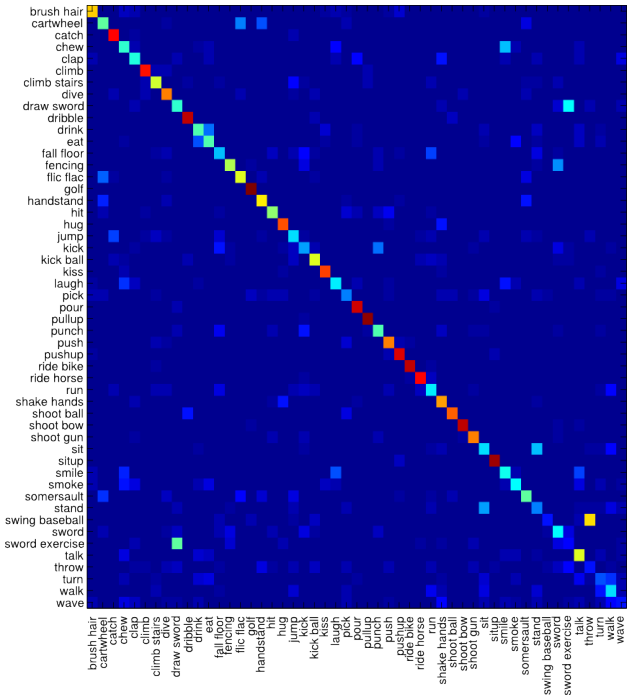

Figure 5: Confusion matrix for the action sequences in the HMDB dataset [35] using the approach as described in one of our previous studies (Husain et al. [7]).

5. Conclusions

We introduced the basic idea behind Convolutional Neural Networks for the task of semantic labeling. We discussed different ways to further enhance the segmentation results by extracting different features from the scene such as the *distance-from-wall*. The semantic labeling can serve as a useful prior for object discovery methods as shown in one of our previous studies in [42].

We also explained the basic approach for recognizing actions in videos using Convolutional Neural Networks and different ways to bring robustness.

References

[1] A. Dragan, N. Ratliff, S. Srinivasa, Manipulation planning with goal sets using constrained trajectory optimization, in: Int. Conf. on Robotics and Automation (ICRA), 4582–4588, 2011.

[2] D. Martínez, G. Alenyà, C. Torras, Planning robot manipulation to clean planar surfaces, Engineering Applications of Artificial Intelligence (EAAI) 39 (2015) 23–32.

[3] F. Husain, A. Colome, B. Dellen, G. Alenya, C. Torras, Realtime tracking and grasping of a moving object from range video, in: Int. Conf. on Robotics and Automation (ICRA), 2617–2622, 2014.

[4] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in: Int. Conf. on Computer Vision (ICCV), 2650–2658, 2015.

[5] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Computer Vision and Pattern Recognition (CVPR), Conf. on, 3431–3440, 2015.

[6] F. Husain, H. Schulz, B. Dellen, C. Torras, S. Behnke, Combining Semantic and Geometric Features for Object Class Segmentation of Indoor Scenes, IEEE Robotics and Automation Letters (RA-L) 2 (1) (2016) 49–55.

[7] F. Husain, B. Dellen, C. Torras, Action Recognition based on Efficient Deep Feature Learning in the Spatio-Temporal Domain, IEEE Robotics and Automation Letters (RA-L) 1 (2) (2016) 984–991.

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems (NIPS), 1097–1105, 2012.

[9] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, CoRR abs/1409.1556.

[10] J. Dai, K. He, J. Sun, BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation, in: The IEEE International Conference on Computer Vision (ICCV), 2015.

[11] G. Lin, C. Shen, A. van dan Hengel, I. Reid, Efficient piecewise training of deep structured models for semantic segmentation, in: Computer Vision and Pattern Recognition (CVPR), Conf. on, 2016.

[12] C. Farabet, C. Couprie, L. Najman, Y. Lecun, Scene parsing with multiscale feature learning, purity trees, and optimal covers, in: Int. Conf. on Machine Learning (ICML), ACM, New York, NY, USA, 575–582, 2012.

[13] H. Noh, S. Hong, B. Han, Learning Deconvolution Network for Semantic Segmentation, in: Int. Conf. on Computer Vision (ICCV), 2015.

[14] S. Hong, H. Noh, B. Han, Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation, in: Advances in Neural Information Processing Systems (NIPS), 1495–1503, 2015.

[15] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: ICCV Workshop on 3D Representation and Recognition, 2011.

[16] H. Schulz, N. Höft, S. Behnke, Depth and height aware semantic RGB-D perception with convolutional neural networks, in: Europ. Conf. on Neural Networks (ESANN), 2015.

[17] S. Gupta, R. Girshick, P. Arbelaez, J. Malik, Learning rich features from RGB-D images for object detection and segmentation, in: Europ. Conf. on Computer Vision (ECCV), vol. 8695, 345–360, 2014.

[18] C. Couprie, C. Farabet, L. Najman, Y. LeCun, Indoor semantic segmentation using depth information, in: Int. Conf. on Learning Representations (ICLR), 1–8, 2013.

[19] S. Khan, M. Bennamoun, F. Sohel, R. Togneri, Geometry driven semantic labeling of indoor scenes, in: Europ. Conf. on Computer Vision (ECCV), 679–694, 2014.

[20] J. Stückler, B. Waldvogel, H. Schulz, S. Behnke, Dense real-time mapping of object-class semantics from RGB-D video, Journal of Real-Time Image Processing (2015) 599–609.

[21] A. Müller, S. Behnke, Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images, in: Int. Conf. on Robotics and Automation (ICRA), 6232–6237, 2014.

[22] D. Wolf, J. Prankl, M. Vincze, Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters, in: Int. Conf. on Robotics and Automation (ICRA), 4867–4873, 2015.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) Challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[24] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in: Europ. Conf. on Computer Vision (ECCV), 746–760, 2012.

[25] S. Song, S. Lichtenberg, J. Xiao, SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite, in: Computer Vision and Pattern Recognition (CVPR), Conf. on, 2015.

[26] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks, in: CVPR, Columbus, OH, United States, 2014.

[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, in: ICML, 2014.

[28] K. Simonyan, A. Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, in: NIPS, Curran Associates, Inc., 568–576, 2014.

[29] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale Video Classification with Convolutional Neural Networks, in: CVPR, 1725–1732, 2014.

[30] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks, in: ICCV, 4489–4497, 2015.

[31] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, in: Computer Vision and Pattern Recognition (CVPR), Conf. on, 2015.

[32] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised Learning of Video Representations using LSTMs, in: Int. Conf. on Machine Learning (ICML), 2015.

[33] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, S. Gould, Dynamic Image Networks for Action Recognition, in: Computer Vision and Pattern Recognition (CVPR), Conf. on, 2016.

[34] K. Soomro, A. R. Zamir, M. Shah, UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild, CoRR abs/1212.0402, URL `http://arxiv.org/abs/1212.0402`.

[35] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: ICCV, 2556–2563, 2011.

[36] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, K. Saenko, Long-term recurrent convolutional networks for visual recognition and description, in: CVPR, 2625–2634, 2015.

[37] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised Learning of Video Representations using LSTMs, CoRR abs/1502.04681.

[38] O. R. Murthy, R. Goecke, Combined ordered and improved trajectories for large scale human action recognition, in: ICCV Workshop on Action Recognition with a Large Number of Classes, 2013.

[39] S. Karaman, , L. Seidenari, A. Bagdanov, A. Bimbo, L1-regularized Logistic Regression Stacking and Transductive CRF Smoothing for Action Recognition in Video, in: ICCV Workshop on Action Recognition with a Large Number of Classes, 2013.

[40] H. Wang, C. Schmid, Action Recognition with Improved Trajectories, in: ICCV, 3551–3558, 2013.

[41] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A Robust and Efficient Video Representation for Action Recognition, International Journal of Computer Vision (2015) 1–20ISSN 0920-5691.

[42] G. Martín García, F. Husain, H. Schulz, S. Frintrop, C. Torras, S. Behnke, Semantic Segmentation Priors for Object Discovery, in: Int. Conf. on Pattern Recognition (ICPR), accepted, 2016.