Action Recognition Based on Efficient Deep Feature Learning in the Spatio-Temporal Domain

Farzad Husain, Babette Dellen, and Carme Torras

***Abstract*—Hand-crafted feature functions are usually designed based on the domain knowledge of a presumably controlled environment and often fail to generalize, as the statistics of real-world data cannot always be modeled correctly. Data-driven feature learning methods, on the other hand, have emerged as an alternative that often generalize better in uncontrolled environments. We present a simple, yet robust, 2-D convolutional neural network extended to a concatenated 3-D network that learns to extract features from the spatio-temporal domain of raw video data. The resulting network model is used for content-based recognition of videos. Relying on a 2-D convolutional neural network allows us to exploit a pretrained network as a descriptor that yielded the best results on the largest and challenging ILSVRC-2014 dataset. Experimental results on commonly used benchmarking video datasets demonstrate that our results are state-of-the-art in terms of accuracy and computational time without requiring any preprocessing (e.g., optic flow) or a priori knowledge on data capture (e.g., camera motion estimation), which makes it more general and flexible than other approaches. Our implementation is made available.**

***Index Terms*—Computer vision for automation, recognition, visual learning.**

I. Introduction

BUILDING personal robots for tasks involving assistance and interaction with humans carries several challenges. One key challenge is to perceive and interpret dynamic human environments. This is necessary for the active engagement of the robot. Several attempts have been made to address the different perception aspects of such dynamic environments where the robot is meant to assist, such as tracking a hand-held object for grasping [1], capturing human motion [2], activity recognition [3] and sensing the human behaviors [4].

One important objective is the detection and recognition of daily human activities. Actions such as brushing hair, eating, drinking, chewing, sitting, walking, standing, etc., implicitly encompass the structure of a particular human environment. Successful recognition of these actions simplifies several tasks that are aimed for such robotic assistants. For example, assisting the elderly in timely caregiving [5], [6], in the situation of accidents [7] or in the daily life activities [8].

Recognizing human activities for robots is conventionally tackled using a pipeline approach, by first (i) modeling the dynamics of changing environments using a graphical model [9]–[11] or identifying descriptive features [12]–[15], and then (ii) performing classification [16], [17]. The first part requires extraction of motion information through some mechanism. Possible approaches include the computation of optic flow or motion modeling. However, recent benchmarks have revealed that there is no universally accepted model that could outperform others for all datasets [18]. The reason is that the statistics of datasets can be considerably different, and a particular model might perform better for one dataset than for another. Many spatio-temporal descriptors are extensions from single image descriptors such as SIFT3D [17], HOG3D [12] and SURF3D [19]. However, such extensions also inherit the limitations in performance generalization as shown in [20], making clear the advantage of learned features over hand-crafted ones.

Deep Convolutional Neural Networks (CNNs) [21] have emerged as a state-of-the-art solution for a wide range of computer vision problems, such as image segmentation/labeling [22], [23], object detection/localization [24] and pose recovery [25], [26]. The main advantage over the conventional pipeline approaches is that CNNs can be trained end-to-end (from raw pixels to labels) in a fully supervised way. One drawback of fully supervised deep learning is that it requires a huge number of labeled training examples [27].

Recently, it has been shown that a CNN model trained from a large dataset can be transferred to other visual recognition tasks with limited training data and thereby leading to higher accuracy and shorter training period [24], [28]. Since single-image input-based models that have been trained over a million labeled images are now readily available [29], we see attempts to exploit these networks in the video domain [30], [31]. However we observe limited success when learning directly in the temporal domain.

We also notice that weakly annotated video data is becoming prevalent as time goes by. For example, 300 hours of video are uploaded to Youtube every minute[1]. Such abundance of video data opens up the opportunity to exploit the infinite space of possible actions in the context of human action recognition [31]. Visual recognition methods have to interpret video data displaying a large degree of variability and complexity in order to arrive at a semantic description, i.e., the action class of the recorded scene.

We propose to recognize human actions using the transfer learning technique. A pretrained single-image recognition model is adapted for videos by temporally concatenating the

Manuscript received August 31, 2015; accepted January 28, 2016. Date of publication February 12, 2016; date of current version March 8, 2016. This paper was recommended for publication by Associate Editor S. Gould and Editor J. Kosecka upon evaluation of the reviewers' comments. This work was supported by the CSIC project TextilRob (201550E028), and in part by the project RobInstruct (TIN2014-58178-R).

F. Husain and C. Torras are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 08028 Barcelona, Spain (e-mail: shusain@iri.upc.edu; torras@iri.upc.edu).

B. Dellen is with the RheinAhrCampus der Hochschule Koblenz, 53424 Remagen, Germany (e-mail: dellen@hs-koblenz.de).

Digital Object Identifier 10.1109/LRA.2016.2529686

[1] https://www.youtube.com/yt/press/statistics.html

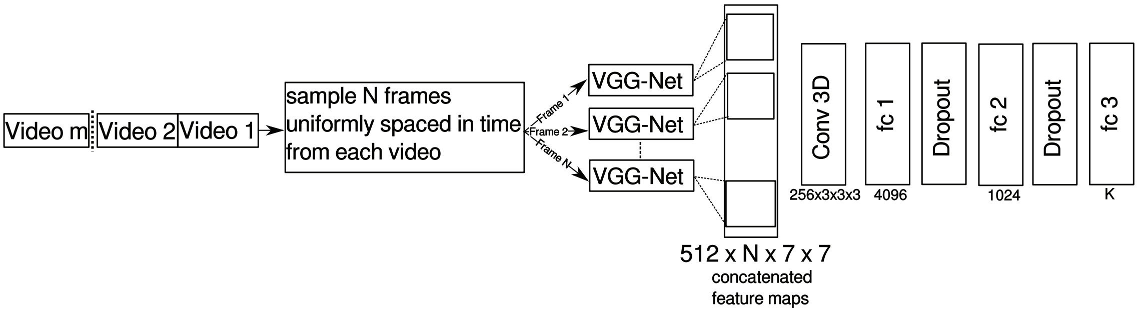

Fig. 1. Illustration of the network. We use the output from layer 16 of the VGG-Net (Table 1 in [45]), as a descriptor. The output is concatenated to form 512, 3D feature maps. The 3D feature maps are used as input for the network consisting of a volumetric convolutional layer followed by two fully-connected layers.

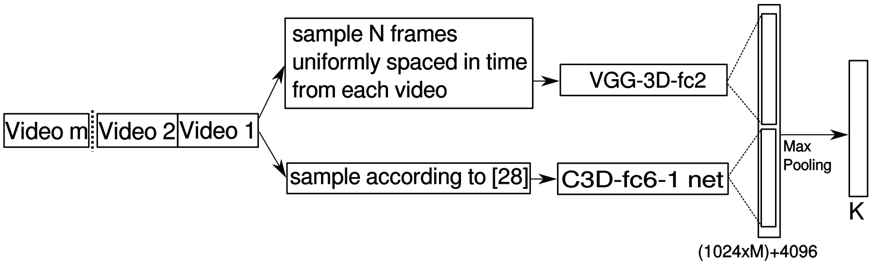

Fig. 2. Illustration of how the different network outputs are combined, where VGG-3D-fc2 refers to the fc2 layer in Fig. 1.

output of its deepest spatial convolution layer. The input to the model are the individual frames of the video. Afterwards, the concatenated output is used as an input to a network comprising 3D convolutions that we train.

The feature representation becomes more abstract as we go deeper in a network thereby obscuring the locally occurring temporal changes in a video. This poses a limitation to the temporal features that the network learns from the concatenated output. We overcome this limitation by combining the output of our learned network with another pretrained model [32] which employs 3D convolutions from the beginning. The complementary nature of the two features becomes evident from the improved recognition accuracy in our experiments.

The combined output contains fewer trainable parameters thereby allowing us to use a more efficient optimization method (L-BFGS) [33]. Our model does not require any pre-computation of features such as optic flow or any other domain-specific processing, thereby making it generic and computationally efficient.

Our main contributions are:

- the introduction of a concatenation scheme in the temporal domain to extend the usage of pretrained models learned from a single image to the video domain,
- combining our learned network with another action recognition model, which yields improved results as compared to the individual networks, and
- evaluation and comparison with commonly used benchmarking video datasets.

II. Related Work

Several action recognition methods have been proposed in the past. We roughly group them into two categories. First is the conventional pipeline approach (descriptor followed by a classifier) [12], [17], [19], [34]–[36] and second is the convolutional model [20], [31], [37]–[39] which is the basis of our approach.

TABLE I
Average Accuracy on the UCF-101 Dataset (3-fold)

Algorithm	Accuracy
CNN with transfer learning [31]	65.4%
LRCN (RGB) [49]	71.1%
Spatial stream ConvNet [30]	72.6%
LSTM composite model [50]	75.8%
Our approach (VGG-3D)	**79.1%**
C3D (1 net) [32]	82.3%
Temporal stream ConvNet [30]	83.7%
C3D (3 nets) [32]	85.2%
Combined ordered and improved trajectories [51]	85.4%
Stacking classifiers and CRF smoothing [52]	85.7%
Improved dense trajectories [34]	85.9%
Improved dense trajectories with human detection[53]	86.0%
Our approach (VGG-3D + C3D-fc6-1 net)	**86.7%**
Spatial and temporal stream fusion [30]	88.0%

In [34], improved dense trajectories are produced by reducing the camera motion effect, which is estimated using the SURF descriptor [40]. However, for recognizing human actions, inconsistent matches from SURF are removed by exploiting domain knowledge, i.e., by adding a human detector. A higher-level representation of activities, named as "action bank," combined with a linear SVM classifier is proposed in [35].

Another way of representing actions is through spatio-temporal segmentation of dynamic scenes. The segmented surfaces and how they change over time gives cues about the kind of manipulation actions which are being carried out. The manipulation actions can be encoded in the form of "Semantic Event Chains" [36]. These event chains represent the spatial relations between objects in the scene. Any change in the spatial

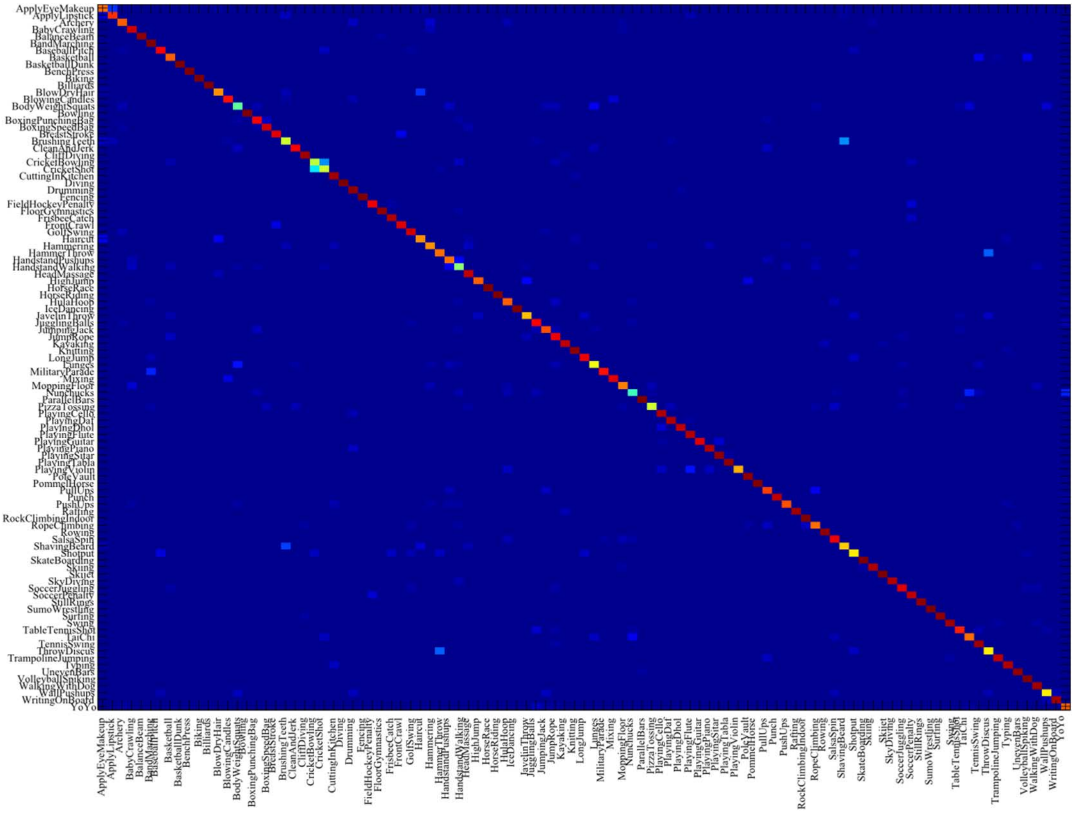

Fig. 3. Confusion matrix for the UCF-101 dataset accumulated for all three splits.

relation serves as a decisive key point through which a manipulation could be defined. Similarly, temporal segmentation of a video into multiple events is proposed in [41].

An unsupervised learning method based on convolution and stacking has also been proposed [20]. The convolved output of arbitrarily sized videos is made constant by dimensionality reduction using principal component analysis. The time-efficient dimensionality reduction for long video clips has a relatively larger memory requirement (up to 32 GB). In [37], a spatio-temporal sparse auto-encoder is trained in an unsupervised way for classifying video sequences. The convolutional gated Restricted Boltzmann Machine architecture [42] has been extended to 3D to learn relations between pairs of adjacent images in videos and is used to extract features for activity recognition [38].

A 3D CNN model has also been previously proposed [39]. In this model, features are learned simultaneously in the spatial and temporal dimensions by performing 3D convolutions. The model is applied to real-world environments to recognize human actions. However, other than the raw images, a set of hardwired kernels is created to generate the gradients and optic flow which should be learned by the proposed convolutional network. In addition, a human detector is introduced and foreground extraction is performed. On the contrary, we feed the raw image data directly to our network and do not compute any handcrafted feature.

In [30], a two-stream network is proposed, where each frame of the video is used as an individual image during training. One stream is trained on raw images and the other is trained with optical flow fields computed from the consecutive video frames. Recognition is attained using a score aggregation strategy across all the video frames of both streams.

Using pretrained models is also proposed in [31], [32], [43]. Our model could be categorized as the *late fusion model* from [31], in which multiple networks are fused in the final fully connected layers. However, we use a different network architecture which is pretrained on a single-image database instead of a video database, thereby reducing the computational cost. We get significantly better results without requiring fine tuning of the pretrained models. Along the same lines, different aggregation strategies for per frame image-based features is investigated in [43].

Our approach is closely related to [32]. A 3D convnet is defined with convolutional kernels up to the first 8 layers. However, we use 3D convolutional kernels after extracting the output from a very deep pretrained model and thereby learn features in the temporal domain at a higher level of abstraction.

III. Problem Formulation

Given a set of videos, obtain a label for each video characterizing its content. The video can be of arbitrary spatial and temporal dimensions.

IV. Approach

We use a single-image convolution model for individual frames of video data and perform volumetric convolution at a higher level of abstraction by temporally concatenating the output. The method is illustrated in Fig. 1. Here, K is the number of action categories. In this way we are able to

TABLE II
CONVNET ACCURACY UNDER DIFFERENT SETTINGS FOR UCF-101 DATASET

Scenario	Accuracy
Fine tune top 3 layers (Sports 1M - pretrained) [31]	65.4% (3 fold)
Fine tune all layers (Sports 1M - pretrained) [31]	62.2% (3 fold)
Spatial AlexNet-stream (pretrained and last layer) [30]	72.7% (1 fold)
Spatial AlexNet-stream (pretrained and fine tuned) [30]	72.8% (1 fold)
Spatial VGG-stream (pretrained and fine tuned)	71.4% (1 fold)
VGG-3D (pretrained and fine tuned)	75.5% (1 fold)
Spatial VGG-stream (pretrained and adaptation layers)	76.3% (1 fold)
VGG-3D (pretrained and adaptation layers)	80.0% (1 fold)
VGG-3D (pretrained and fine tuned) + C3D-fc6-1 net	83.5% (1 fold)
VGG-3D (pretrained and adaptation layers) + C3D-fc7-1 net	84.8% (1 fold)
VGG-3D (pretrained and adaptation layers) + C3D-fc6-1 net	86.7% (1 fold)

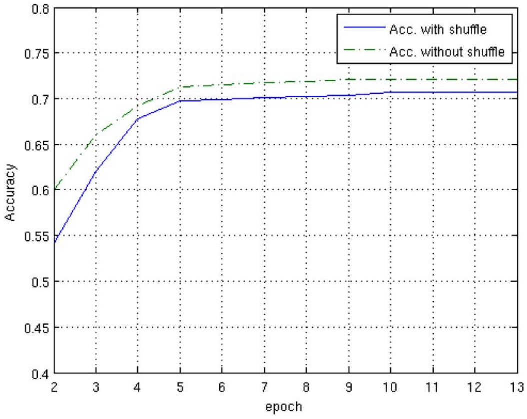

Fig. 4. Comparing accuracy for shuffling video sample frames.

TABLE III
AVERAGE ACCURACY ON THE HMDB DATASET (3-FOLD)

Algorithm	Accuracy
Spatio-temporal HMAX network [54]	22.8%
Spatial stream ConvNet [30]	40.5%
Trajectory-Based Modeling [55]	40.7%
Our approach (VGG-3D)	**46.9%**
Decomposing visual motion [56]	52.1%
Our approach (VGG-3D + C3D-fc6-1 net)	**53.9%**
Temporal stream ConvNet [30]	54.6%
Improved dense trajectories [34]	57.2%
Spatial and temporal stream fusion [30]	59.4%

initialize our network with the parameters learned from the ImageNet dataset [44]. Additionally, we freeze the learned network parameters up to the second-last fully-connected layer, combine the output with another pretrained network and build a new softmax model. We refer to the CNN architecture (19-layer network in [45]) as VGG-Net.

A. Feature Map Concatenation

We train a network which takes as input 3D feature maps. These feature maps are the outputs from layer-16 of the 19-layer network defined in [45] and trained on the ImageNet dataset. Layer-16 is the last spatial convolution layer in [45]. This gives us a high-level feature descriptor in the spatial domain. Afterwards we add one 3D convolutional layer followed by three fully-connected layers. The K-way softmax function is applied to the output of the last fully-connected layer, where K is again the number of action categories. In total, our network contains 20 layers and we train only the last 4 layers. We use a dropout regularization ratio of 0.5 for the fully-connected layers.

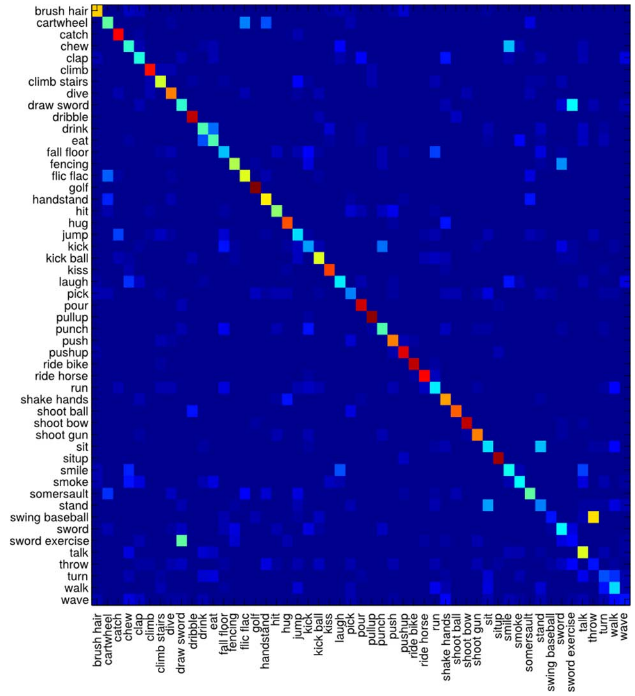

Fig. 5. Confusion matrix for the HMDB dataset accumulated for all three splits.

We take $N = 30$ uniformly spaced frames from each video as input to the network. Each image is assigned the same label as its corresponding video. The network is trained using stochastic gradient descent and use the same momentum as in [44]. The learning rate is adjusted to get the maximum accuracy in a minimum number of iterations on a held-out validation set from the training set.

B. Combining Multiple Networks

A deep network learns different features at each level of the layer hierarchy. The activations in the initial layers tend to be more sensitive to edge-like patterns and corners within their receptive field, whereas activations at deeper levels have larger receptive fields and capture more complex invariances [46].

Our network learns changes that occur in the temporal domain at a more abstract level because we temporally concatenate the output of the convolutional layer-16 from the pretrained network of [45]. Hence, our model lacks learning in the temporal domain from locally occurring changes. In order to palliate this deficiency, we concatenate the fully connected layer-9 feature vectors with a length of 4096, extracted from another deep network that was trained in 3D from the beginning [32]. The model contains 8 3D-convolution, 5 max-pooling, and 2 fully-connected layers. Deeper 3D convolution layers are not possible, due to GPU memory restrictions. The model is trained on the Sports-1M dataset [31] which contains about 1 million videos of different sports action categories.

Figure 2 shows the combination scheme of the two feature maps. We concatenate the output of the fully-connected layers for the same video, hence having the same action category. We also augment the data by cropping $M = 10$ patches from each

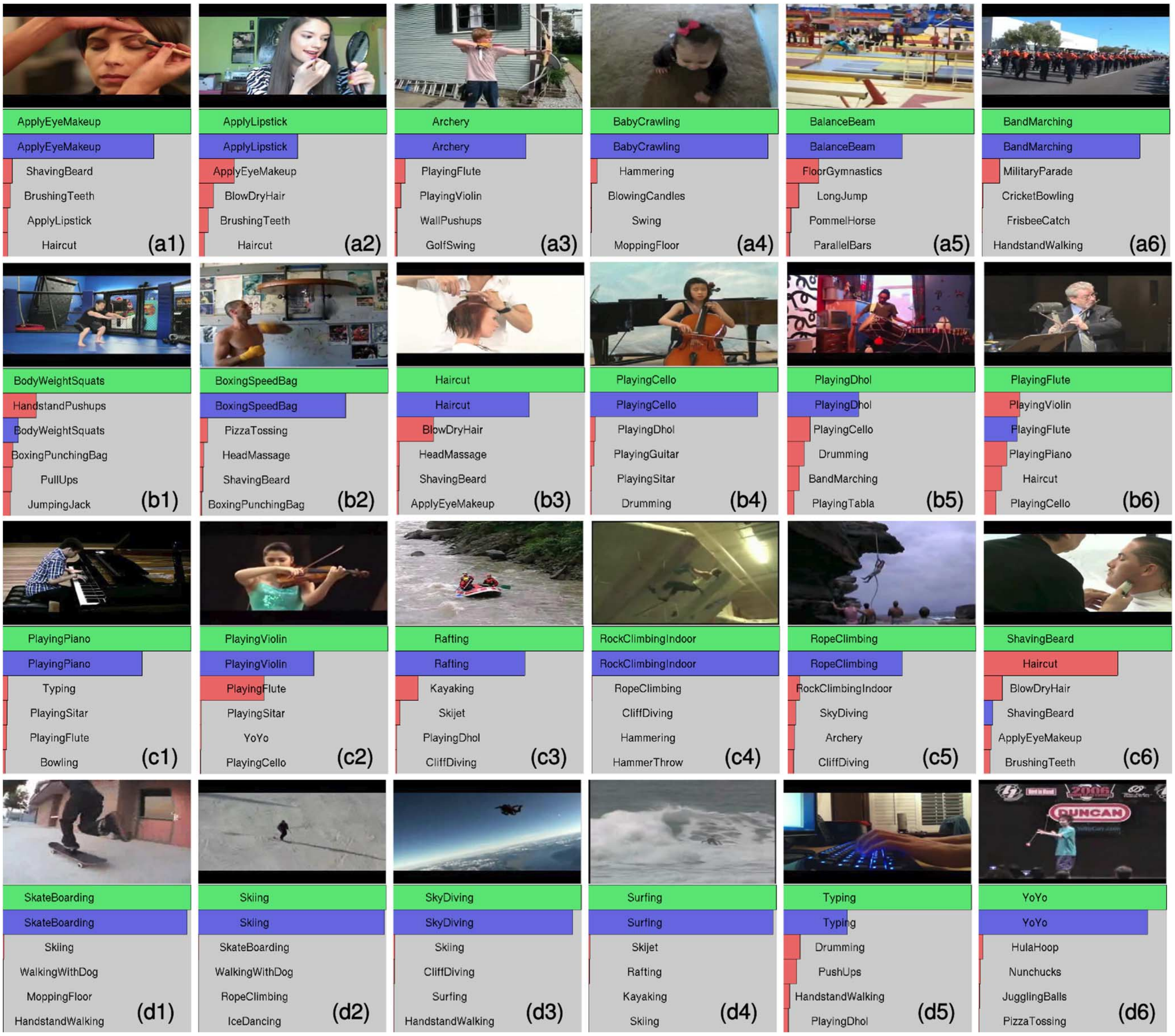

Fig. 6. Top-5 predictions using our approach for selected test sequences from the UCF-101 dataset [47] with 101 action categories. First row (green color) shows the ground-truth followed by predictions in decreasing level of confidence. Blue and red show correct and incorrect predictions, respectively.

frame of the video for the VGG-3D network, hence the output feature dimension is $4096 + (1024 \times M)$. After concatenation, we perform max-pooling to reduce the feature dimension and afterwards build a new softmax model. Since we are learning the parameters for the softmax layer only, we can use a more efficient optimization approach instead of stochastic gradient descent. We use an off-the-shelf implementation[2] of L-BFGS which has been shown to yield better results when the number of trainable parameters is small [33].

V. Experiments

We evaluate our approach on two publicly available benchmarking datasets, UCF-101 [47] and HMDB [48]. These datasets are challenging because many video samples include camera motion as well as a dynamic background. We use the same evaluation protocol as proposed by the respective authors and provide an in-depth analysis of our approach using the UCF-101 dataset as a test case. Additional qualitative results for both the datasets are available at http://www.iri.upc.edu/people/shusain/actionrecognition.html

The network is able to take only fixed-size input frames, hence we resize all the videos so that the maximum dimension is 256 pixels and crop 10 patches of size 224 × 224 pixels according to the data augmentation scheme as proposed in [44]. Furthermore, we separate 10% percent of the samples from the training data and use them as validation data. Such data are needed to determine the number of iterations needed for stochastic gradient descent.

A. UCF-101 Dataset

The UCF-101 [47] dataset contains 13,320 labeled video samples with 101 action categories. We use the 3-way train/test

[2] http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html

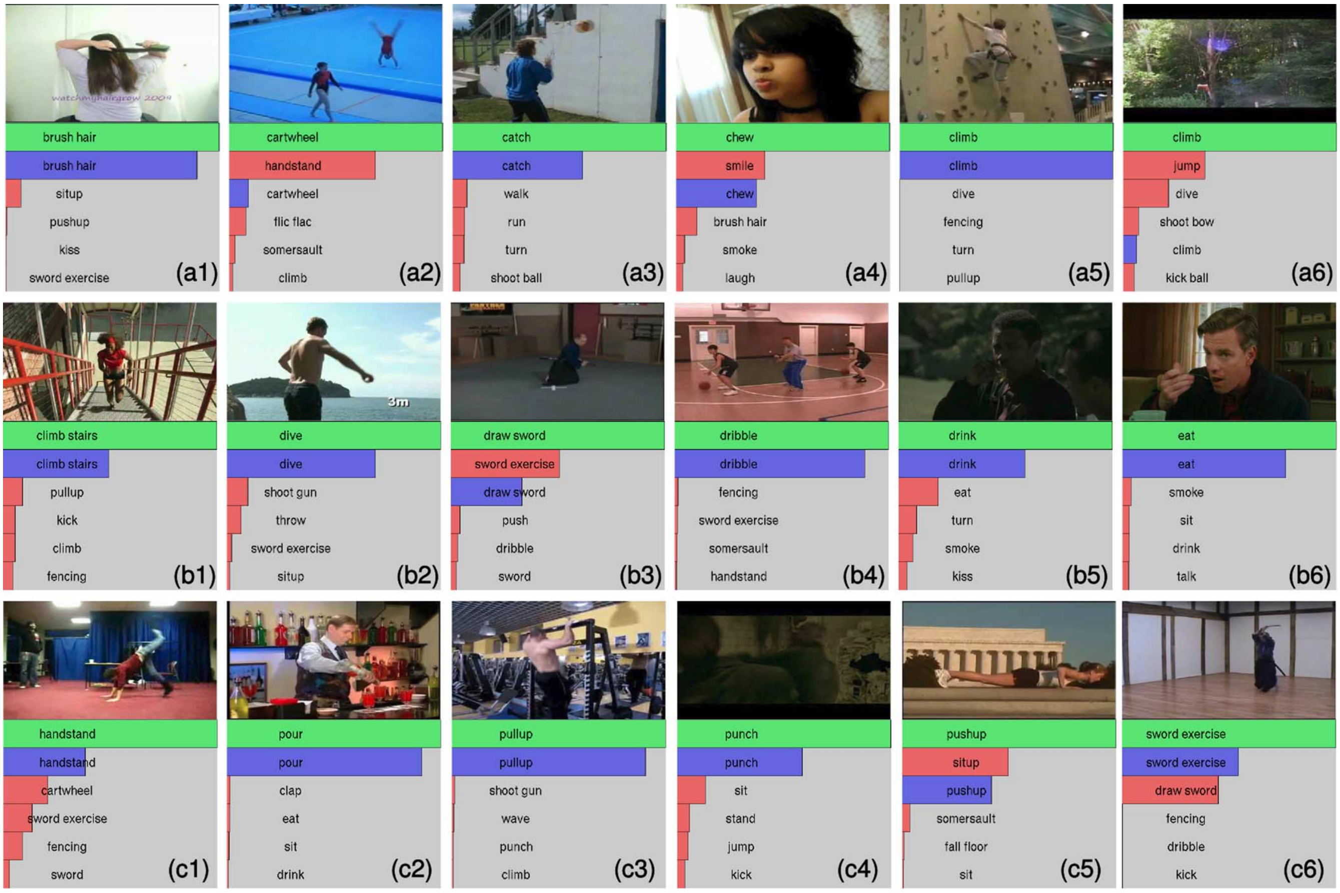

Fig. 7. Top-5 predictions using our approach for selected test sequences from the HMDB dataset [48] with 51 action categories. First row (green color) shows the ground-truth followed by predictions in decreasing level of confidence. Blue and red show correct and incorrect predictions, respectively.

split as provided by the authors. Table I shows a comparison with other approaches. Compared with the baseline [31] we observe a considerable improvement. Not surprisingly, we see improved results as also compared to [32]. This shows the complementary nature of the high-level (layer-19) and low-level (layer-6) features. It should be noted that we use the output from the layer-9 activation (C3D 1 net) and concatenate it with our trained model as described in Fig. 2, i.e., concatenating two networks, as opposed to [32], where the output from 3 networks that have been trained differently is combined. Our results are closer to [30], where optical flow needs to be computed. However, the calculation of optical flow leads to a significant computational overhead. As shown in [32], Brox optical flow used in [30] takes 0.85-0.95s per image pair which is 274x slower than C3D. Additionally, storing the raw flow fields for this dataset requires a disk space of 1.5 TB which needs data compression [30]. Figure 3 shows the confusion matrix accumulated for all the three splits. Comparing the confusion matrix with that resulting from the approach in [30] (Fig. 5 in [30]), it can be seen that the actions "CricketBowling" and "CricketShot" have similar levels of confusion, whereas our approach shows better results for the action "YoYo". Figure 6 shows the top-5 predictions for selected test sequences from the UCF-101 dataset [47] with 101 action categories.

B. Evaluating Different Scenarios

We measure the performance of our spatial and spatio-temporal learning framework in different scenarios using the split-1 of UCF-101 dataset. Table II presents the evaluation under different settings along with the comparison to other approaches.

In our spatial VGG-stream we obtained the label for a video after averaging the scores for all the frames belonging to that video. All the layers were pretrained on the ImageNet dataset and fine tuned on the UCF-101 dataset, except the last layer which was initialized randomly because of different number of classes. We observed results similar to the spatial AlexNet-stream [30]. We found better results for spatial VGG-stream and VGG-3D when training only the adaptation, i.e., the newly added layers. Similar behavior was observed in [31], i.e., a drop in the accuracy when fine tuning all the layers. This is because training such a huge network with a small dataset results in overfitting. We observed the best result when training the adaptation layers only combined with the fc6 layer from C3D.

C. Learning From Temporal Information

Due to the concatenation of the feature maps in the temporal domain, the 3D kernels should also be able to exploit the temporal information in the video. Hence, if we randomly shuffle the video frames while training, we should see a drop in the accuracy due to temporal inconsistency. Figure 4 shows the drop in the accuracy averaged for two training sessions of the method described in Sec. IV-A for split-1 of UCF-101 dataset.

D. HMDB Dataset

The HMDB dataset [48] contains 6,849 labeled video samples with 51 action categories. We use the 3-way train/test split as provided by the authors. Table III shows a comparison with other approaches. The methods from [30] and [34] perform better than ours, however, both require computation of dense per frame optical flow for each video. In addition, the method in [34] also requires camera motion estimation. Figure 5 shows the confusion matrix accumulated for all three splits. It can be seen that similar actions such as “throw” and “swing baseball” are the most confused. Figure 7 shows the top-5 predictions for selected test sequences.

E. Qualitative Analysis

Since we do not preprocess the data using techniques such as background subtraction or tracking a bounding box, our feature-learning approach is agnostic to such domain-specific information. For this reason, wrong labels can be seen, in Figs. 6 and 7, when different activities are performed in visually similar environments. For example, Fig. 6(c6) vs. Fig. 6(b3) and Fig. 6(b6) vs. Fig. 6(c2), share similar environments and we see a high confidence of “HairCut” in the “ShavingBeard” action and “PlayingFlute” got confused with “PlayingViolin”. Similar observations can be made in Fig. 7(b3) vs. Fig. 7(c6). However, sometimes background plays an important role in correctly recognizing certain actions, for instance, “SkyDiving” (Fig. 6(e3)) and “Surfing” (Fig. 6(e4)).

Other than similar background, actions may themselves be also visually confusing, which can affect feature learning. For example, Fig. 7(a2) vs. Fig. 7(c1). Both activities “cartwheel” and “handstand” entail performing a similar motion.

These are inherent problems of feature learning when using only raw data and the resulting mislabelings have been named *reasonable mistakes* in [31].

VI. Conclusions

We tackled the problem of action recognition by using a spatio-temporal feature learning scheme. This scheme allowed us to exploit the record-breaking pretrained single image classification model [45]. We report extensive experimental evaluations using challenging action recognition datasets. Our results are competitive with the state-of-the-art convolutional and strong feature-based baselines.

We are using the publicly available Torch7 library for our implementation which is optimized for fast processing on a CPU as well as a GPU. In our timing experiments we found that for a batch size of 60 videos, it takes ~ 1.6 seconds on a modern GPU to perform a single forward and backward pass through the network in Fig. 1 and about 6 hours (14 epochs) to train on complete UCF-101 training set. We expect to get further speed up by a more efficient implementation of 3D convolution.

So far, we concatenated the feature maps in the last convolutional layer. In the future, we plan to explore possible modifications in the network design to further exploit learning in the temporal domain. One possibility would be to gradually increase the number of temporal connections along the sequence of layers. This kind of adaptation is proposed in [31]. We also plan to investigate the effect on performance of gradually clipping the top layers of the network and evaluation on the recently introduced Sports-1M dataset [31] which contains over 1 million labeled sample videos.

References

[1] F. Husain, A. Colome, B. Dellen, G. Alenya, and C. Torras, “Realtime tracking and grasping of a moving object from range video,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2014, pp. 2617–2622.

[2] J. Chan and G. Nejat, “A learning-based control architecture for an assistive robot providing social engagement during cognitively stimulating activities,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2011, pp. 3928–3933.

[3] A. Chrungoo, S. Manimaran, and B. Ravindran, “Activity recognition for natural human robot interaction,” in *Social Robotics*, vol. 8755, Australia: Springer International Publishing, 2014, pp. 84–94.

[4] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, “A communication robot in a shopping mall,” *IEEE Trans. Rob.*, vol. 26, no. 5, pp. 897–913, Oct. 2010.

[5] R. Liu, X. Zhang, J. Webb, and S. Li, “Context-specific intention awareness through web query in robotic caregiving,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2015, pp. 1962–1967.

[6] C. Schroeter *et al.*, “Realization and user evaluation of a companion robot for people with mild cognitive impairments,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2013, pp. 1153–1159.

[7] S. Nakagawa *et al.*, “Tandem stance avoidance using adaptive and asymmetric admittance control for fall prevention,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2015, pp. 5898–5903.

[8] K.-H. Park, H.-E. Lee, Y. Kim, and Z. Bien, “A steward robot for human-friendly human-machine interaction in a smart house environment,” *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 21–25, Jan. 2008.

[9] H. Koppula, R. Gupta, and A. Saxena, “Learning human activities and object affordances from RGB-D videos,” *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 951–970, 2013.

[10] N. Hu, G. Englebienne, Z. Lou, and B. Krose, “Learning latent structure for activity recognition,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2014, pp. 1048–1053.

[11] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Unstructured human activity detection from RGBD images,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2012, pp. 842–849.

[12] A. Klaeser, M. Marszalek, and C. Schmid, “A spatio-temporal descriptor based on 3D-gradients,” in *Proc. 19th Br. Mach. Vis. Conf. (BMVC)*, 2008, pp. 99.1–99.10.

[13] C. S. Stefan Mathe, “Dynamic eye movement datasets and learnt saliency models for visual action recognition,” in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, 2012, vol. 7573, pp. 842–856.

[14] I. Laptev, “On space-time interest points,” *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[15] H. Wang, H. Zhou, and A. Finn, “Discriminative dictionary learning via shared latent structure for object recognition and activity recognition,” in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2014, pp. 6299–6304.

[16] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local svm approach,” in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 2004, vol. 3, pp. 32–36.

[17] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 357–360.

[18] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *Proc. Br. Mach. Vis. Conf. (BMVC)*, 2009, pp. 124.1–124.11.

[19] G. Willems, T. Tuytelaars, and L. Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 650–663.

[20] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2011, pp. 3361–3368.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[23] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 82–90.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Nov. 2014, pp. 1717–1724.

[25] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2337–2344.

[26] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "Modeep: A deep learning framework using motion features for human pose estimation," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2014, pp. 302–315.

[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," arXiv preprint arXiv:1406.2227, 2014.

[28] J. Donahue *et al.*, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, vol. 32, pp. 1–9.

[29] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.

[30] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 568–576.

[31] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1725–1732.

[32] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4489–4497.

[33] Q. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Ng, "On optimization methods for deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 265–272.

[34] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 3551–3558.

[35] S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1234–1241.

[36] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation," *Int. J. Rob. Res.*, vol. 30, no. 10, pp. 1229–1249, Sep. 2011.

[37] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification," in *Proc. Br. Mach. Vis. Conf. (BMVC)*, 2012, pp. 124.1–124.1.

[38] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 140–153.

[39] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[40] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Comput, Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[41] H. Zhang, W. Zhou, and L. Parker, "Fuzzy segmentation and recognition of continuous human activities," in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, 2014, pp. 6305–6312.

[42] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.

[43] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," in *Proc. Br. Mach. Vis. Conf. (BMVC)*, 2015, pp. 60.1–60.13.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Int. Conf. Learning Representations (ICLR)*, vol. abs/1409.1556, 2015, pp. 1–14.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 818–833.

[47] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012, [Online]. Available: http://arxiv.org/abs/1212.0402

[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2556–2563.

[49] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2015, pp. 2625–2634.

[50] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," *Proc. 32nd Int. Conf. Mach. Learning (ICML)*, vol. abs/1502.04681, 2015, pp. 843–852.

[51] O. R. Murthy and R. Goecke, "Combined ordered and improved trajectories for large scale human action recognition," in *Proc. ICCV Workshop Action Recognit. Large Number Classes*, 2013, pp. 412–419.

[52] S. Karaman, L. Seidenari, A. Bagdanov, and A. Bimbo, "L1-regularized logistic regression stacking and transductive CRF smoothing for action recognition in video," in *Proc. ICCV Workshop Action Recognit. Large Number Classes*, 2013, pp. 1–6.

[53] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 1, pp. 1–20, 2015.

[54] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2007, pp. 1–8.

[55] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, vol. 7576, pp. 425–438.

[56] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proc. IEEE Conf. Comput, Vis. Pattern Recognit. (CVPR)*, 2013, pp. 2555–2562.