Human Interaction Prediction Using Deep Temporal Features

Qiuhong Ke[1(✉)], Mohammed Bennamoun[1], Senjian An[1], Farid Boussaid[2], and Ferdous Sohel[3]

[1] School of Computer Science and Software Engineering, The University of Western Australia, Crawley, Australia
qiuhong.ke@research.uwa.edu.au,
{mohammed.bennamoun,senjian.an,farid.boussaid}@uwa.edu.au
[2] School of Electrical, Electronic and Computer Engineering, The University of Western Australia, Crawley, Australia
[3] School of Engineering and Information Technology, Murdoch University, Murdoch, Australia
F.Sohel@murdoch.edu.au

Abstract. Interaction prediction has a wide range of applications such as robot controlling and prevention of dangerous events. In this paper, we introduce a new method to capture deep temporal information in videos for human interaction prediction. We propose to use flow coding images to represent the low-level motion information in videos and extract deep temporal features using a deep convolutional neural network architecture. We tested our method on the UT-Interaction dataset and the challenging TV human interaction dataset, and demonstrated the advantages of the proposed deep temporal features based on flow coding images. The proposed method, though using only the temporal information, outperforms the state of the art methods for human interaction prediction.

Keywords: Interaction prediction · CNN · Temporal convolution

1 Introduction

Interaction prediction, or early event recognition, aims to infer an interaction at its early stage [1]. It can help in preventing harmful events (*e.g.*, fighting) in a surveillance scenario. It is also essential to robot-human interaction (*e.g.*, when a human lifts his/her hand or opens his/her arms, the robot could then respond accordingly).

Unlike interaction recognition, interaction prediction requires the inference of the action before it occurs. This requires the prediction of any potential future action, using the frames captured prior to the action. We can see from Fig. 1 that it is difficult to infer the action class from a single frame. The temporal information and the combination of several frames, on the other hand, provide more information about the future action class. In this paper, we focus on the

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part II, LNCS 9914, pp. 403–414, 2016.
DOI: 10.1007/978-3-319-48881-3_28

Fig. 1. Human interaction prediction. The goal is to predict the interaction class before it happens, which is difficult to achieve from a single frame.

temporal information of video sequences and introduce a new deep temporal feature for human interaction prediction.

Existing interaction prediction methods mainly use spatial features (*e.g.*, bag-of-words) [1], or combine spatial and temporal features (*e.g.*, histogram of oriented optical flow) [2] to represent the video frames. These hand-crafted features are, however, not powerful enough to capture the salient motion information for interaction prediction due to their loss of the global structure in the data [3]. Recent works in large-scale recognition tasks [4,5] show that deep learned representations perform better than the traditional hand-crafted features. The generic features extracted with the pre-trained convolutional neural networks (CNN) are very powerful for image classification and object detection tasks [6–9]. In this paper, we show that the same pre-trained CNN model can also be used to extract deep temporal features. This is possible since a deep neural network can learn transferable features for a wide variety of vision tasks [9]. The CNN models are trained with natural images, so in order to extract effective representations, the key step is to represent the temporal information (optical flow) in a manner that is compatible with natural images. We propose to use the flow coding images as a low-level temporal information and extract deep temporal features from them using a CNN model. We show that the proposed deep temporal features outperform the methods which combine low-level spatial and temporal representations. In addition, we propose to learn convolutional filters to combine consecutive video frames. Specifically, we investigate two learning convolution methods: simultaneous convolution and separate convolution. We show (Sect. 3) that the learning convolutional filters further improves the accuracy compared to a simple average pooling.

The main contributions of this paper include: **(1)** the introduction of flow coding images to represent the low-level temporal information of a video sequence; **(2)** the extraction of deep temporal features using a pre-trained CNN model from ImageNet [10] and the learning of temporal convolution across frames; **(3)** extensive experiments show that the proposed method, though using only the temporal information, outperforms the state of the art methods which combine spatial and temporal features [2].

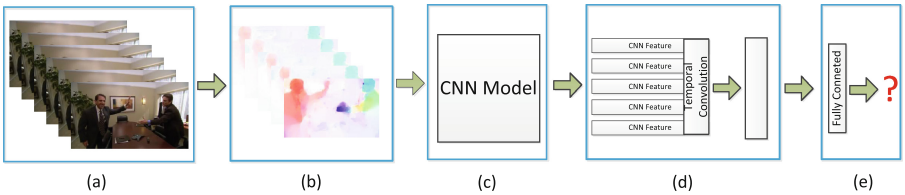

Fig. 2. Outline of our approach. (a) Sample frames from the TV human interaction dataset [11]. (b) The flow coding images computed from the input frames are normalized by cropping the human regions. (c) Deep temporal CNN feature extraction. The publicly available pre-trained CNN-M-2048 model [12] is used to extract CNN features. (d) Temporal convolution learning. The CNN features are concatenated to learn the temporal convolution. (e) Fully connected feedforward neural network, including a hidden layer to reduce the feature dimension and a softmax layer to output the interaction class.

2 Proposed Approach

A work flow of our method is shown in Fig. 2. It includes four parts. **First**, flow coding images are computed from consecutive video frames. **Second**, a deep CNN network is used to extract deep temporal features from flow coding images. **Then**, the output features of several frames are concatenated together to learn the temporal convolution. **Finally**, a fully connected feedforward neural network, including a hidden layer and a softmax layer, is used for classification. Next, we describe the details of these four steps.

2.1 Flow Coding Image

In this paper, we use the optical flow between two consecutive frames to represent the motion information. The most widely used techniques for optical flow computation are differential methods [14]. These algorithms are based on the assumptions of constancy of the intensity (i.e., the grey values of two consecutive frames do not change over time) and the smoothness of flow field (i.e., the total variation of the flow field should not be too large) [15].

Let v_x and v_y be the x and y components of the optical flow at a pixel, respectively. The flow coding image is obtained as follows [16,17]. First, the magnitude r and the angle θ of the velocity are computed:

$$r = \sqrt{v_x^2 + v_y^2)} \tag{1}$$

$$\theta = \arctan 2\left(\frac{v_y}{v_x}\right) + \pi \tag{2}$$

The next step computes the color for each velocity according to the magnitude and angle of the velocity. The idea is based on the color wheel [18]. As shown in Fig. 3, different hues are assigned to different orientations, varying from

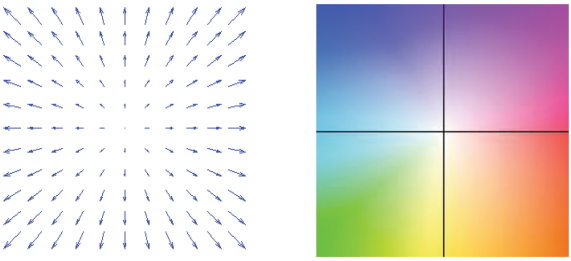

Fig. 3. Color coding of flow vector. Each flow vector is assigned a color according to the orientation and magnitude of the vector. The hue of the color represents the orientation while the saturation represents the magnitude. (Color figure online)

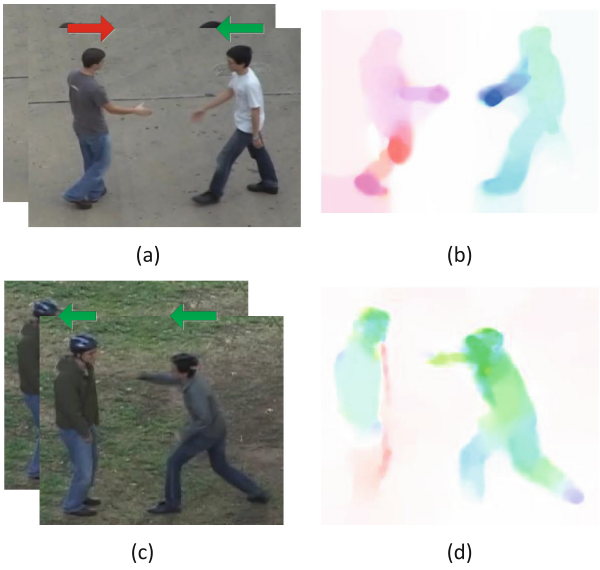

Fig. 4. Flow coding image. (a) and (c) are video frames from the UT-Interaction dataset Set 1 and Set 2 [13]. The arrows show the overall motion directions of the humans. (b) and (d) are the corresponding flow coding images. The orientations and magnitudes of the flow vectors are mapped to the corresponding hues and saturations of colors. (Color figure online)

red, yellow, green, cyan, blue to magenta. As the magnitude becomes larger, the saturation increases. Two examples of flow coding images are shown in Fig. 4. Compared with the raw video frames, the flow coding images retain the human motion in a more explicit way. It can be seen that the humans with different directions of motions are assigned different colors, with variant saturation corresponding to different magnitudes of the motions. In addition, the flow coding images assign the white color to the background as there is no motion and the

magnitude is zero. Therefore, the flow coding images do not contain complex (*e.g.*, textured) backgrounds, especially when the video is captured with a static camera.

2.2 Deep Temporal CNN Feature

CNN Model. Once the flow coding images are obtained from the optical flow, they are fed into the pre-trained CNN-M-2048 model [12] to extract deep temporal features. The network was learnt on ILSVRC-2012 [10]. The architecture is shown in Fig. 5. It is similar to the one used by Zeiler and Fergus [8]. In the convolutional layers, there are 96 to 1024 kernels with size varying from 3×3 to 7×7. The rectification (ReLU) [19] is used as a nonlinear activation function. For robustness to intra-class deformations, max pooling kernels of size 3×3 with stride 2 are used at different layers. In all experiments, the output of the first fully connected layer (layer 19) of the network is used as the feature vector.

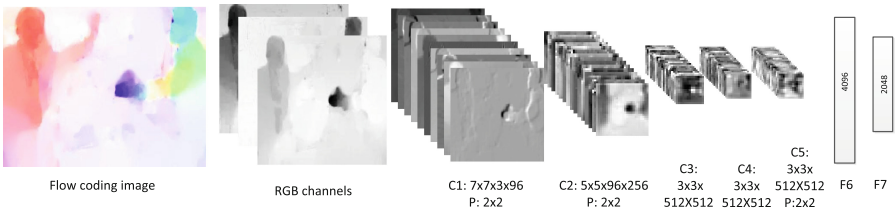

Fig. 5. Single CNN feature extraction. The input is the flow coding image (i.e., a single frame). The architecture is based on CNN-M-2048 model. "C", "P" and "F" denote "convolution", "max pooling" and "fully connected", respectively. The output of "F7" is used as the temporal feature vector.

Region of Interest. Human interaction like handshake or kicking involves the stretching of hands or legs. These body parts are important for accurate interaction prediction. However, the bounding boxes of humans do not include all of the body parts. In this case, the region of interest (ROI) is selected by merging the bounding boxes of the two humans. Each flow coding image is then normalized by cropping the ROI. As shown in Fig. 6, for the TV human interaction dataset, the given upper body annotations are used to select the left, right and upper bound of the ROI. The lower bound is chosen as the height of the frame. For the UT-Interaction dataset, the tracking algorithm [20] is applied to detect humans. The ROI is selected as the region $[x_{min} : x_{max}, y_{min} : y_{max}]$, where x_{min}, x_{max}, y_{min}, y_{min} denote the minimum and maximum x and y coordinates of the human bounding boxes, respectively.

2.3 Learning Temporal Convolution & Classification

As shown in Fig. 6, our ROI covers the two interacting humans. We investigate two different methods to extract CNN features and learn the temporal convolution using the ROI:

(1) **Simultaneous convolution:** the ROI is fed into the deep model as a whole image to extract CNN features. For each frame, the dimension of the output feature vector is 2048. This method tries to learn the temporal convolution for two humans simultaneously.
(2) **Separate convolution:** the ROI is divided into two equal sized images: the left and the right half images. The CNN features are extracted separately from these half images. They are then concatenated as a large feature vector. As a result, the dimension of the output feature vector is 4096 for each flow coding image. This method separates the two interacting humans and learns the convolution for each of them separately.

The output deep temporal features of several consecutive frames are concatenated in the temporal dimension to form a *temporal image* (as shown in Fig. 2(c)). A fully temporal convolution is applied to combine several consecutive frames. This fully temporal convolution is used to compute the weights of the consecutive frames. Given a *temporal image* generated from a sequence of several frames, a 1D temporal convolutional filter is learned. The size of the filter is the same as the number of frames. The vector can be treated as the weighted sum feature vector of these features, as shown in Eq. (3).

$$\mathbf{x_i} = \left[\, \mathbf{v}_{i-k+1}\ \mathbf{v}_{i-k+2} \cdots \mathbf{v}_i \,\right] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \tag{3}$$

where $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_k]^T$ denotes the learning convolutional filter. $\mathbf{v}_j$ denotes the CNN feature at frame j. k is the number of concatenated frames. $\mathbf{x}_i$ denotes the output feature vector after convolution, which is fed to a "ReLU" transform function. The output is further reduced dimension with a full connected (FC) layer:

$$\mathbf{z} = f(W_1^T f(\mathbf{x}) + \mathbf{b_1}) \tag{4}$$

where $\mathbf{z}$ is the output of the first hidden layer, W_1 and $\mathbf{b_1}$ are the weight matrix and the bias of the first hidden layer, respectively. $f(\cdot)$ is the "ReLU" activation function given by:

$$f(\mathbf{x}) = \max(0, \mathbf{x}) \tag{5}$$

$\mathbf{z}$ is fed to another FC layer with a softmax activation function to output the final probability distribution of classes (i.e., y):

$$\mathbf{y} = g(W_2^T \mathbf{z} + \mathbf{b_2}) \tag{6}$$

where W_2 and $\mathbf{b_2}$ are the model weight and the bias of the output layer, respectively. $g(\cdot)$ is the softmax operation function given by:

$$\begin{aligned} g(\mathbf{z}) &= [g_1(\mathbf{z}), g_2(\mathbf{z}), \cdots, g_d(\mathbf{z})] \\ g(z_j) &= \frac{e^{z_j}}{\sum_{i=1}^{d} e^{z_i}}, \quad j = 1, \cdots, d \end{aligned} \tag{7}$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_d]$, d is the dimension of z. In this case, d is the number of action classes. $\mathbf{y}$ is the probability distribution of classes. The loss function between the probability distribution and the ground-truth class label is computed as follows:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\mathbf{y} \left(c_i \right) \right) \tag{8}$$

where N is the number of training samples, c_i is the class of the ith data. The model parameters are updated using the mini-batch stochastic gradient descent algorithm [21].

Fig. 6. Region of interest (ROI). (a) Video frame from the TV human interaction dataset. The green box is the provided upper body annotation. The red box is the selected ROI. (b) and (c) are video frames from the UT-Interaction dataset Set 1 and Set 2, respectively. The green box is the detected human bounding box. The red box is the selected ROI. Each flow coding image is cropped into the ROI before feeding it to the CNN model. (Color figure online)

3 Experiments

We tested our method on the TV human interaction dataset [11] and the UT-Interaction dataset [13]. The network weights were learnt using the mini-batch stochastic gradient descent with a batch size of 100. Each 100 training data was uniformly sampled across the class. Based on empirical test, the optimal number of nodes in the hidden layer was set to 512. The momentum was set to 0.9.

3.1 TV Human Interaction Dataset

This dataset consists of 300 video clips collected from more than 20 different TV shows. It contains 5 interaction classes: hand shake, high five, hug, kiss and a "none" class that dose not contain any of the interactions above, such as talking and walking. The dataset also provides annotations for each frame of the video, including the bounding boxes of the upper human bodies, action labels for each person and whether there is interaction or not. As mentioned in Sect. 2.2, in each frame, we used the provided bounding boxes of humans in actions to extract the ROI.

We used the training/testing split provided along with the dataset and we trained all of the frames from the training videos. The training epoch was set to 30. The learning rate was initialized to 10^{-3}, and was decreased to 10^{-4} after 30 epochs.

The testing protocols in [2] were adopted. The prediction accuracy was tested on 5 different temporal stages (i.e., distance between the testing frame and the starting frame of the interaction, measured in the number of frames), from –20 to 0, with a step size of 5. "−20" denotes that the temporal distance between the testing frame and the starting frame of the interaction is within 20 frames. "0" means that the testing frames are within 5 frames after the interaction happened. We tested the accuracies of learning simultaneous convolution and separate convolution with a convolution filter size of 3 and 7. The comparison results are shown in Fig. 7. Our simultaneous convolution learning method outperforms the state of the art. The accuracy is about 5 % better in the "–10" temporal stage, and it is 15 % better in the final temporal stage. The results show that our deep temporal features are much more powerful in capturing the motion information

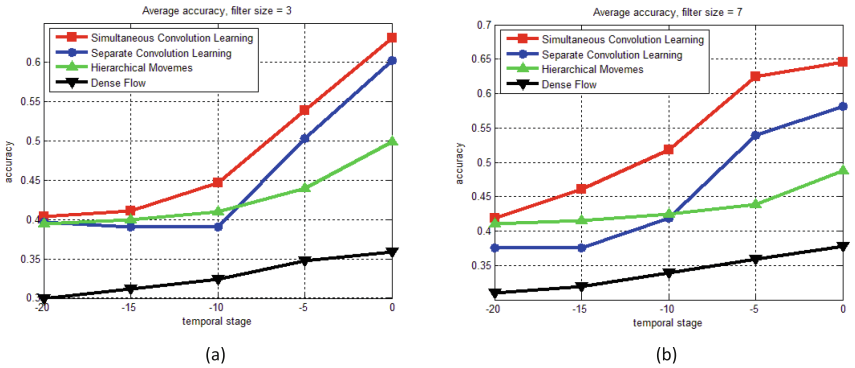

Fig. 7. Early interaction prediction accuracies on the TV human interaction dataset. (a) and (b) show comparisons of our two convolution learning methods and comparisons with other methods, including the "hierarchical movemes" and dense flow [2] on the TV human interaction dataset. Our simultaneous convolution learning method outperforms the state of the art.

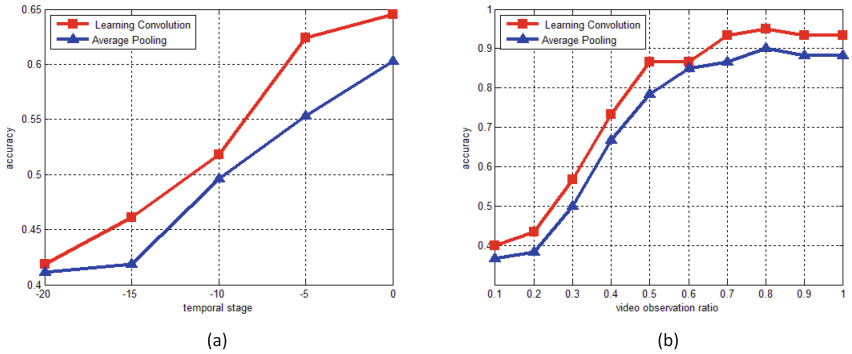

Fig. 8. Performance comparison of the learning convolution and the average pooling method: (a) TV human interaction dataset; (b) UT-interaction dataset.

in the video sequences and perform better than the combination of low-level spatial and temporal hand-crafted features.

For the TV interaction dataset, the simultaneous convolution performs better than the separate convolution. This is because the TV show dataset comes from movies and contains a lot of shots from different viewpoints with moving cameras and various viewpoints. Some frames include only one person. Learning the convolution simultaneously is more robust than separating the frames into two halves.

We also compared the performances of learning simultaneous convolution and the average pooling given the CNN deep features extracted from the flow coding images of 7 frames. Both methods output one feature vector which is fed into neural network for classification. The results are shown in Fig. 8(a). The learning temporal convolution performs better than the simple average pooling. The accuracy is about 8 % better in the "−5" temporal stage. The results show that using learning convolution to produce the weighted sum of consecutive frames captures more salient motion information in the video sequence than a simple average pooling.

3.2 UT-Interaction Dataset

This dataset includes two sets. In Set 1 the background is simpler and mostly static. In contrast, in Set 2 the background is complex and slightly moving. Each set includes 10 sequences. Each sequence is segmented into 6 classes of videos, i.e., handshake, hug, pointing, kick, push and punch.

There are no training/testing splits provided with this dataset. The performance is measured using leave-one-sequence-out cross validation, i.e., for each set, 9 sequences of segmented videos (54) are used for training and the remaining 6 videos are used for testing. The learning rate was set to 10^{-3}, and the training stopped after 30 epochs.

Table 1. Recognition performance with respect to the first half length of the videos. The accuracy of most previous methods drops dramatically on Set 2 where the background is more complicated and the set is more challenging compared to Set 1. Our method achieves state-of-the-art recognition performance when only half of the videos are available, and outperforms the current best result [2] by about 12 % on Set 2.

Methods	Accuracy wrt. half videos (Set 1)	Accuracy wrt. half videos (Set 2)
Our Method	**88.3 %**	**81.7 %**
hierarchical movemes [2]	83.1 %	69.0 %
Dynamic BoW [1]	70.0 %	54.0 %
Integral BoW [1]	61.8 %	49.4 %
Cuboid + Bayesian [1]	30.0 %	28.0 %
Cuboid + SVMs [1]	31.7 %	35.0 %
BP + SVM [22]	65.0 %	54.0 %

We learned the separate convolution with a filter size of 5 to recognize 50 % of videos (i.e., given a video of d frames, using the sequence of $[1, round(0.5 * d)]$ to predict the action class). The results are shown in Table 1. Our method outperforms the state of the art. The accuracy of most previous methods drops significantly on Set 2 where, again, the background is more complicated and challenging. Our method achieves state-of-the-art prediction when only the half of the video is available, and outperforms the current best result [2] by about 5 % when tested on Set 1 and about 12 % when tested on Set 2.

Figure 9 illustrates the details of interaction prediction when tested on Set 2. Each video was tested with 10 different observation ratios from 0.1 to 1, with a step of 0.1, *e.g.*, given the video length d, the observation ratio 0.3 means that the sequence of $[1, round(0.3 * d)]$ is provided for testing. We tested our method with learning simultaneous convolution and separate convolution with a convolution filter size k of 3 and 7. Given a test video of n frames, we derive $d-k+1$ predicting classes. The prediction of the video is given by the max voting strategy, i.e., the number of each class is calculated in the $d - k + 1$ classes and the maximum number of class is chosen as the interaction class of the video. We can see that our separate convolution learning method outperforms the state of the art. It improves the accuracy by roughly 9 % when only the first 10 % of the length of each video is provided. When the full length of the videos is provided, the improvement is about 10 %. It shows that the proposed method is valuable for both prediction and recognition. The separate convolution is seen to perform better than the simultaneous convolution. For the UT-Interaction dataset, given that the actions are captured with a static view point, the separate convolution can learn specific and detailed temporal information for each of the two humans.

Figure 8(b) compares the learning of the separate convolution and the average pooling given the deep temporal features extracted from flow coding images of 7 frames. We can see that learning temporal convolution performs better than the average pooling method.

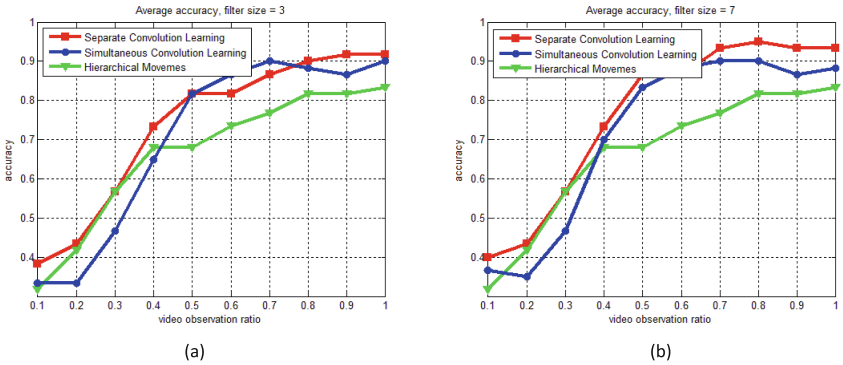

Fig. 9. Early interaction prediction accuracies on the UT-Interaction dataset. The comparisons of our two convolution learning methods and the "hierarchical movemes" approach [2] on the UT-Interaction dataset. The video observation ratio denotes the given testing sequence length, *e.g.*, "0.3" means that the testing sequence consists of the first 30 % of the video frame. Our separate convolution learning method performs better than the "hierarchical movemes" approach under different observation ratios.

4 Conclusion

In this paper, we have proposed a learning method to exploit the deep temporal features of video frames for human interaction prediction. We tested our method on the UT-Interaction dataset and the challenging TV human interaction dataset. Although our method only uses temporal information, it still outperforms the state of the art when tested on challenging datasets. The experimental results clearly illustrate that the proposed learning method enables an early recognition of human interaction.

Acknowledgement. This work was partially supported by Australian Research Council grants DP150100294, DP110103336, and DE120102960.

References

1. Ryoo, M.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1036–1043. IEEE (2011)
2. Lan, T., Chen, T.-C., Savarese, S.: A hierarchical representation for future action prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part III. LNCS, vol. 8691, pp. 689–704. Springer, Heidelberg (2014)
3. Acar, E., Hopfgartner, F., Albayrak, S.: Understanding affective content of music videos through learned representations. In: Gurrin, C., Hopfgartner, F., Hurst, W., Johansen, H., Lee, H., O'Connor, N. (eds.) MMM 2014, Part I. LNCS, vol. 8325, pp. 303–314. Springer, Heidelberg (2014)
4. 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: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3361–3368. IEEE (2011)

5. Ren, X., Ramanan, D.: Histograms of sparse codes for object detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3246–3253. IEEE (2013)
6. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)
7. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1717–1724. IEEE (2014)
8. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014)
9. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 512–519. IEEE (2014)
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis., 1–42 (2014)
11. Patron-Perez, A., Marszalek, M., Zisserman, A., Reid, I.D.: High five: Recognising human interactions in tv shows. In: BMVC, vol. 1, p. 2, Citeseer (2010)
12. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
13. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: 2009 IEEE 12th International Conference on Computer vision, pp. 1593–1600. IEEE (2009)
14. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. Int. J. Comput. Vision **61**(3), 211–231 (2005)
15. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
16. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vision **92**(1), 1–31 (2011)
17. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Citeseer (2009)
18. http://members.shaw.ca/quadibloc/other/colint.htm. Accessed 15 June 2015
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
20. Ess, A., Leibe, B., Schindler, K., Gool, L.V.: A mobile vision system for robust multi-person tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008. pp. 1–8. IEEE (2008)
21. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT 2010, pp. 177–186 (2010)
22. Laviers, K., Sukthankar, G., Aha, D.W., Molineaux, M., Darken, C., et al.: Improving offensive performance through opponent modeling. In: AIIDE (2009)