Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 471–477

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos

J. Arunnehru[a,*], G. Chamundeeswari[a], S. Prasanna Bharathi[b]

[a] *Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus, Chennai - 26.*
[b] *Department of ECE, SRM Institute of Science and Technology, Vadapalani Campus, Chennai - 26.*

Abstract

In recent days, suspicious action recognition is a significant topic in intelligent video surveillance and computer vision research. Action recognition methodologies are specially needed for surveillance systems which are required to prevent crimes and treacherous actions before occurring. In this paper, we present 3D - Convolutional Neural Networks (3D-CNN) with 3D motion cuboid for action detection and recognizing in videos. The experiments are conducted on benchmark KTH and Weizmann dataset. The proposed method is compared with the existing methods in terms of accuracy. The results show that this approach is outperforms previously published results.

Keywords: Human action recognition; 3D Convolutional neural network; 3D motion information; Temporal difference; Classification

1. Introduction

In modern days, recognizing human action or activity in public places is a significant problem in the area of video surveillance and computer imaging. However, it is a big challenge to identify the different activity of human precisely in various environmental conditions, whereas many scenarios, consists of insignificant data in the video such as messy backgrounds or various point of view, can interrupt our analytic system and diminish the precision of recognition. In the current year, there is a number investigation about identifying the human action based on the complications like human motions which can be branched into four zones: human gestures, human actions, human interactions and group activities. This work focused on the human action classification and recognition. There are several approaches using the various dataset to solve action recognition problem with the certain hypothesis, which does not give a good practical value for the real-time environment. Initially, the specific measure is used to represent the features from the extracted raw video frames in order to identify the human actions. Where these assumptions will fail to give information in

* Corresponding author. Tel.: +91 7010727200
E-mail address: arunnehru.j@vdp.srmuniv.ac.in

10.1016/j.procs.2018.07.059

ending and it is difficult to identify the specific feature in the actual scenario. The above problem can be solved partially by Deep Learning models which follow a multi-layered with space-time approach by feeding the input training data, which can learn the features and compute the results from the particular datasets without making any assumptions.

This paper focuses on identifying human actions more accurately based on the space-time motion information cuboid and modeled using 3D Convolutional Neural Networks (CNNs).

2. Related work

CNNs come under the group of biologically influenced models for identification of visual information. HMAX [1] is a prototype which has been advanced for recognizing the observed object on inspired by the arrangement of visual cortex in human. In this paper, some previous works [2, 3, 4] are motivated to solve the real time challenges in action recognition problems. In the HMAX system, a grouping of progressively going on convoluted features are produced by the diverse applications such as max pooling and template matching. The primitive model of HMAX is mainly designed to inspect the two-dimensional images which have been extensively used in recognizing actions in the video.

A new method of human action recognition, which applies a scale of Self Organizing Maps along with supervised neural networks approaches based on the locomotion of joints in human is presented in [5]. This paper also involves in processing the inputs captured from the 2D camera and gathers the information available by using the first and second order dynamics. In addition to this joint based prediction of human action recognition, [6] employs a set of features namely Joint Kinetics and Relational features. This involves a particular joint movement is purely based on certain set of kinetic features available along with the human action. Based on the speed, velocity, acceleration, angular velocity, rate of acceleration, angular acceleration, potential energy and kinetic energy measured from the human action is used as a set of kinetic features to train and test the data. Along with these features, horizontal and vertical distance, orientation of sine and cosine, eigen vector and also the direction specified by the link is taken to predict the human action is well discussed with experimental results in this article. It is difficult to recognize the human action which involves parallel actions along with various factors such as object interaction using deep Convolutional networks with less number of samples.

A new method [7] has been involved using two-stream ConvNets to learn the human actions with increased complexity using less number of sample data. In different pose variations and scattered backgrounds, [8] investigates a framework involving CNN and pose hints to identify the human actions based on the frame work developed for the object recognition using CNNs. At the same time, neglecting the irrelevant information rooted from the object recognition model. In spite of the various approaches available, in order to increase the precision of human action recognition [9] propose a design which involves inserting the motion information present in the human body into depth maps. In this method, along with the movement of human body geometrical arrangement is also considered to recognize the human action, spatial-temporal pyramid is used to compact the specific character and then Simplified Fisher vector encoding method is incorporated to accumulate low level features.

The rest of the paper is organized as follows. Section 3 presents the 3D motion information, Section 4 discuss 3D-Convolutional neural networks. Section 5 presents the Experimental setup and results. Finally, Sect. 6 concludes the paper.

3. 3D - Motion Cuboid

Frame difference approach is immensely adaptive to identify the motion scene similar to moving objects in the dynamic environment. The absolute temporal difference frame is attained by subtracting the earlier frame t with current frame $t+1$ on a pixel by pixel basis. Fig. 1 shows the continuous frames of the 'running' action from KTH dataset. The resulting difference frame stack is called as 3D Motion cuboid. To extract the moving objects from action sequences, frame difference is calculated by Eq. 1.

$$\begin{aligned} &D_t(x,y) = |I_t(x,y) - I_{t+1}(x,y)| \\ &1 \le x \le w, 1 \le y \le h \end{aligned} \tag{1}$$

$$T_k(x,y) = \begin{cases} 1, & \text{if } D_t(x,y) > t; \\ 0, & \text{Otherwise;} \end{cases} \tag{2}$$

where $D_t(x,y)$ is the difference image, $I_t(x,y)$ is the pixel intensity of (x,y) in the t^{th} frame, h and w are the height and width of the video frame correspondingly.

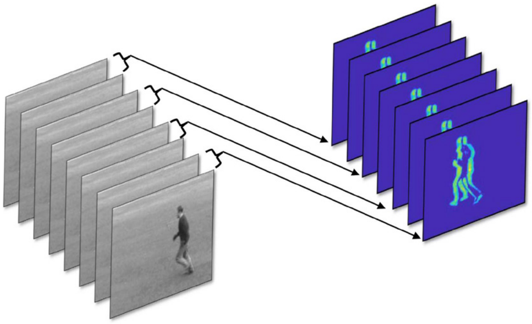

Fig. 1. Extraction of motion information (3D- Motion Cuboid)

4. 3D Convolutional Neural Networks

In two dimensional Convolutional neural networks, convolutions are enforced on the two dimensional maps extracted from the feature to enumerate the features available from the geometrical dimension. We introduce to enumerate three dimensional convolutions in the succeeding stages of CNNs to gauge the features from both the temporal and spatial dimensions. The 3D convolution is obtained by convolving a three dimension kernel to the cube obtained by assembling more than one spatial temporal patches arranged in a contiguous manner. The feature maps present in the convolution layer is linked with the multiple frames arranged contiguously in the previous layer in order to capture the motion related information. It is noted that 3D convolution kernel can select only one type of feature from the patch cuboid, provided the kernel weights are duplicated across the patch cube. A common design scheme of Convolution neural networks is the number of feature maps grows as the layers increases there by developing the various multiple types of features from the available lower level of maps. The 3D convolution is obtained by convolving a 3D filter kernel by stacking multiple contiguous frames together to produce the 3D cube. By this operation, the feature maps are connected to multiple contiguous frames. Formally, the value at position (x,y,z) on the j^{th} feature map in the i^{th} layer is given by

$$v_{i,j}^{x,y,z} = \tanh\left(b_{ij} + \sum_m \sum_{a=0}^{A_i-1} \sum_{b=0}^{B_i-1} \sum_{c=0}^{C_i-1} w_{ijm}^{abc} v_{(i-1)m}^{(x+a)(y+b)(z+c)}\right) \tag{3}$$

where C_i is the 3D filter kernel size along the temporal dimension, w_{ijm}^{abc} is the (a, b, c) is the feature map connected to the m^{th} value of the kernel in the previous layer. The proposed 3D CNN architecture is given in Fig. 2.

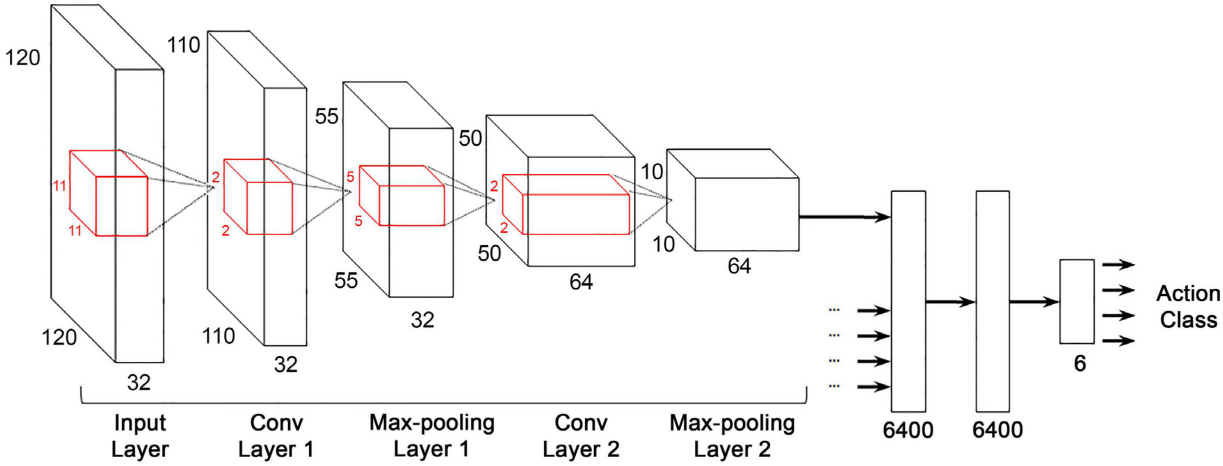

Fig. 2. Overview of the proposed 3D CNN architecture.

To compute the spatial size of 3D CNN output volume using the hyper parameters such as receptive field (R), zero-padding (P), stride length (S) and volume dimension (Width × Height × Depth, or W× H× D). To calculate the Convolutional layer neurons by $((W - F + 2.P)/S) + 1$. The input layer consists of $((120 - 11 + 2.0)/1) + 1 = 110$ which gets output volume of $110 \times 110 \times 32$, here $W, H = 120$ is the height and width of the input frame, $F = 11 \times 11 \times 32$ is the 3D filter depth, $P = 0$ is the zero-padding, and $S = 1$ is the stride leads to the output.

5. Experimental Setup

The analysis is conducted in Python 3.5 with OpenCV 3.1 and TensorFlow in Windows 10 OS on PC with Intel i7 processor with RAM 8 GB. The 3D CNN model is evaluated on the KTH and Weizmann datasets. To succeed the setup in the 3D CNN model, we use an 11-frame cube (motion information) as input. To reduce the computation complexity and memory requirement, the original input frames 160×120 are reduced to 120×120 resolutions in our experiments [10]. 3D CNN architecture is shown in Fig. 2, which consists of $120 \times 120 \times 11$ inputs with the number of feature maps and kernels sizes in each layer. In 3D CNN, the two Convolutional layers use kernel filter of sizes $11 \times 11 \times 32$ and $5 \times 5 \times 32$, respectively, and the two layers of max pooling use kernels size of 2×2 is progressively reduced the spatial which decrease the number of parameters and network computation and also it regulates the overfitting and finally, fully connected layer have all activations in the previous layer which is transformed into 6400 Dimensional feature vectors. The softmax layer consists of output units resultant to the action classes.

5.1. Action Datasets

The KTH video dataset [11] covers six actions such as waving the hands, clapping the hands, running, boxing, walking and jogging executed by 25 different humans in four different scenarios like indoor, outdoor, different scales and clothing appearance. Totally there are around 2931 video data available in the dataset. The videos were captured with 25 frames per second using a static camera and they are down sampled to 160 × 120 spatial resolution. The sample figure depicts the paradigms of datasets. The Weizmann video dataset [12] covers ten different human actions such as walking, running, jumping, galloping sideways, twisting, waving a single hand, skipping, bending, jumping from the place and jumping jack performed by nine different people at outdoor locations. There are totally 90 videos with the resolution of 180 × 144 pixels captured with 25 frames per second using a static camera. The sample frames of the KTH and Weizmann action dataset is shown in the Fig. 3.

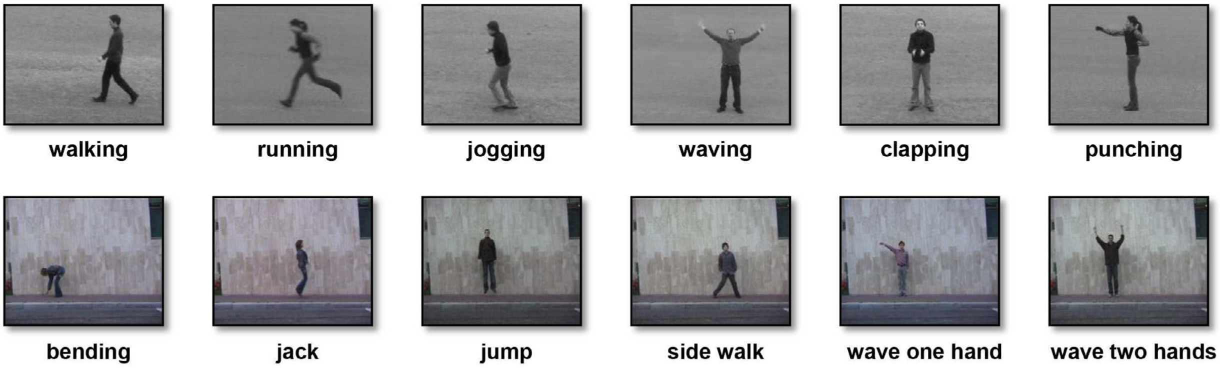

Fig. 3. Sample frames from KTH (top row) and Weizmann (bottom row) action dataset.

5.2. *Experimental Results on KTH dataset*

The confusion matrix of the 3D-CNN classifier on KTH dataset is shown in Table 1, where correct predictions are shown across the table diagonal cells, and most of the actions classes like running, jogging, single hand waving, clapping and boxing are almost predicted well with more than 94%. An average recognition rate of 3D-CNN classifier on KTH dataset is 94.9%. From the confusion matrix, the walking action is misclassified as running due to its similar pattern. Table 2 shows the mean performance of 3D-CNN classifier, which gives good results on precision, recall and F-measure (trade-off between precision and recall).

Table 1. Confusion matrix (%) obtained for the KTH action dataset using 3D CNN

class	walking	jogging	running	boxing	waving	clapping
walking	93.4	1.0	3.3	1.2	0.2	0.8
jogging	0.5	97.4	2.1	0.0	0.0	0.0
running	1.4	2.5	95.4	0.2	0.3	0.2
boxing	1.6	1.0	0.0	94.9	1.8	0.8
waving	1.2	0.2	0.5	3.2	94.0	0.7
clapping	2.1	0.0	0.3	2.3	2.3	93.1

Table 2. Performance metrices obtained for KTH dataset using 3D CNN

class	precision	recall	f-measure
walk	0.932	0.934	0.933
run	0.954	0.974	0.964
jog	0.951	0.954	0.953
wave	0.943	0.949	0.946
clap	0.947	0.940	0.944
box	0.968	0.931	0.949
Average	0.949	0.949	0.949

5.3. Experimental Results on Weizmann dataset

The confusion matrix of the 3D-CNN classifier on Weizmann dataset is shown in Table 3, where correct responses shows in the main diagonal and most of the action classes like walk, side, skip, jump, pjump, jack, bend and wave with one hand are almost predicted well. An average recognition rate of 3D-CNN classifier on Weizmann dataset is 97.2%. From this, wave with both hands action is misclassified as walk and wave with one hand respectively. Table 4 shows the mean performance metrics of 3D-CNN, which gives excellent results on precision, recall and F-measure, where high recall value indicates that CNN classifier returned most of the relevant samples correctly.

Table 3. Confusion matrix (%) obtained for the Weizmann action dataset using 3D CNN

class	walk	run	side	skip	jump	pjump	jack	bend	wave1	wave2
walk	97.9	0.0	0.9	0.2	0.4	0.0	0.4	0.2	0.0	0.0
run	1.1	95.9	0.4	1.1	0.7	0.0	0.7	0.0	0.0	0.0
side	1.2	0.3	97.1	0.5	0.2	0.2	0.5	0.0	0.0	0.0
skip	0.0	0.2	0.0	98.9	0.4	0.0	0.4	0.0	0.0	0.0
jump	0.6	0.2	0.2	0.8	97.9	0.2	0.2	0.0	0.0	0.0
pjump	0.9	0.0	0.2	0.0	0.2	97.0	0.9	0.2	0.5	0.0
jack	0.4	0.1	0.4	0.1	0.3	0.4	98.2	0.0	0.0	0.1
bend	0.4	0.0	0.7	0.0	0.4	1.1	0.7	96.8	0.0	0.0
wave1	1.0	0.0	0.3	0.3	0.3	0.7	0.3	0.7	95.6	0.7
wave2	2.1	0.0	1.1	0.0	1.1	0.5	1.6	0.5	1.6	91.4

Table 4. Performance metrices obtained for Weizamann dataset using 3D CNN

class	precision	recall	f-measure
walk	0.942	0.979	0.960
run	0.981	0.959	0.970
side	0.974	0.971	0.972
skip	0.971	0.989	0.980
jump	0.973	0.979	0.976
pjump	0.975	0.970	0.972
jack	0.974	0.982	0.978
bend	0.982	0.968	0.975
wave1	0.982	0.956	0.969
wave2	0.983	0.914	0.947
Average	0.972	0.972	0.972

5.4. Comparative Study

The results obtained by the 3D-CNN with motion cuboid are compared quantitatively with state-of-the-art results with KTH and Weizmann dataset to measure the effectiveness of the proposed action recognition system and comparison is presented in Table 5. Based on the comparison, it is seen that the proposed method shows best results on KTH and Weizmann action dataset.

Table 5. State-of-the-art recognition accuracies (%) for the KTH and Weizmann

Method	KTH	Weizmann
Proposed	94.9	97.2
Kim et al. [13]	95.33	-
Liu et al. [14]	93.80	91.20
Jhuang et al. [15]	91.70	94.80
Lin et al. [16]	93.43	96.37

6. Conclusion

In this paper, an advanced approach is proposed for suspicious action recognition in intelligent video surveillance. In this work, we used 3D Convolutional neural networks (3D-CNN) with 3D motion cuboid for action detection and recognition in real-time surveillance videos to prevent crimes. The experiments are conducted on KTH and Weizmann dataset. The results exhibits that 3D-CNN outperforms well when compare to state-of-art results.

References

[1] T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 2, Ieee, 2005, pp. 994–1000.

[2] M. K. Geetha, J. Arunnehru, A. Geetha, Early recognition of suspicious activity for crime prevention, Emerging Technologies in Intelligent Applications for Image and Video Processing 205.

[3] J. Arunnehru, M. K. Geetha, Maximum intensity block code for action recognition in video using tree-based classifiers, in: Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Springer, 2015, pp. 715–722.

[4] J. Arunnehru, M. K. Geetha, Motion intensity code for action recognition in video using pca and svm, in: Mining Intelligence and Knowledge Exploration, Springer, 2013, pp. 70–81.

[5] Z. Gharaee, P. Gärdenfors, M. Johnsson, First and second order dynamics in a hierarchical som system for action recognition, Applied Soft Computing.

[6] X. Tian, J. Fan, Joints kinetic and relational features for action recognition, Signal Processing 142 (2018) 412–422.

[7] Y. Han, P. Zhang, T. Zhuo, W. Huang, Y. Zhang, Going deeper with two-stream convnets for action recognition in video surveillance, Pattern Recognition Letters.

[8] T. Qi, Y. Xu, Y. Quan, Y. Wang, H. Ling, Image-based action recognition using hint-enhanced deep neural networks, Neurocomputing 267 (2017) 475–488.

[9] X. Ji, J. Cheng, W. Feng, D. Tao, Skeleton embedded motion body partition for human action recognition using depth sequences, Signal Processing.

[10] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, IEEE transactions on pattern analysis and machine intelligence 35 (1) (2013) 221–231.

[11] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[13] T.-K. Kim, S.-F. Wong, R. Cipolla, Tensor canonical correlation analysis for action classification, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[14] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos âĂIJin the wildâĂİ, in: Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on, IEEE, 2009, pp. 1996–2003.

[15] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, Ieee, 2007, pp. 1–8.

[16] Z. Lin, Z. Jiang, L. S. Davis, Recognizing actions by shape-motion prototype trees, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 444–451.