Pakistan Journal of Life and Social Sciences
www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2025-23.2.00243

RESEARCH ARTICLE

Spatio-Temporal Feature Learning with 3D-CNN For Human-Robot Collaboration Systems

Marina Zhdanova[1*], Viacheslav Voronin[2], Nikolay Gapon[3], Semenishchev Evgeny[4], Zelensky Alexander[5]

[1]Department of Cybersecurity of Information Systems, Don State Technical University, Rostov-on-Don, Russian Federation
[2]Scientific-Manufacturing Complex «Technological Centre»), Zelenograd, Russian Federation
[3]Department of Cybersecurity of Information Systems, Don State Technical University, Rostov-on-Don, Russian Federation
[4]Scientific-Manufacturing Complex «Technological Centre»), Zelenograd, Russian Federation
[5]Scientific-Manufacturing Complex «Technological Centre»), Zelenograd, Russian Federation

ARTICLE INFO

Received: Oct 22, 2025

Accepted: Dec 12, 2025

Keywords

3D-CNN
Human Action Recognition
HRC
S patio-Temporal Features
Video Sequences

***Corresponding Author:**

mpismenskova@mail.ru

ABSTRACT

In this work, an advanced approach to human motion recognition in HUMAN-ROBOT COLLABORATION (HRC) systems is proposed. Classical motion recognition methods, based on manually engineered features, have serious limitations due to their inability to adapt to the unique characteristics of individual performers and limited perception of complex scenes. The introduced approach addresses these limitations by replacing fixed Gabor filters with trainable convolutional layers of a three-dimensional convolutional neural network (3D-CNN), which enables automatic extraction of Spatio-temporal features from video sequences.

1. INTRODUCTION

Current trends in the automation of work processes are characterized by active implementation of robotic technologies across virtually all industrial sectors, including mechanical engineering, agriculture, nuclear power industry, aerospace sector, etc. One of the key development concepts is Human-Robot Collaboration (HRC)—the joint execution of complex tasks by humans and robots within a unified workspace. This concept allows for significant improvement in production efficiency through optimal distribution of functions between an operator and a machine. However, implementing the HRC concept requires effective interaction mechanisms between human operators and robots.

One crucial aspect of this interaction is the ability of robots to accurately recognize operator actions. This becomes particularly important under dynamically changing and heterogeneous working environments typical of modern manufacturing facilities.

To address these challenges, systems capable of processing video data streams from various types of sensors, such as cameras operating outside the visible spectrum, are required. Such systems enable creation of flexible automated environments that facilitate rapid adaptation of production processes to changes in operational conditions.

In the context of HRC, human action recognition is vital for several purposes, including enhancing collaboration between humans and robots [1], optimizing industrial processes [2], assisting operators in performing their duties [3], ensuring occupational health [4], and safeguarding worker safety [5].

Therefore, developing methods for analyzing video streams and algorithms for recognizing operator actions represents a critical scientific problem whose solution will significantly expand the capabilities of robotic systems in collaborating with humans.

2. Related Works

Within the HRC (Human-Robot Collaboration) paradigm, special attention is paid to tracking the movements of the human operator, establishing the context of interaction [6], and predicting subsequent human behavior for completing specific tasks. The context of interaction includes identification of objects (components or tools) with which the operator works, the sequence of actions during task execution, and the conditions of the working environment. Given that the same tasks executed by different operators can differ considerably, the context of interaction exhibits considerable variation. Consequently, accurate recognition of human movements is crucial for building reliable HRC systems [7].

Movement recognition is a critically important element of effective HRC systems [8]. Modern methods for recognizing human actions can be broadly divided into two categories: methods based on computer vision systems and methods relying on wearable sensors. Among these, approaches based on computer vision deserve particular attention because they provide a less invasive method of data collection compared to wearable-sensor-based methods. Typically, these methods rely on depth sensors to construct a three-dimensional skeletal model [2].

Methods for recognizing human actions can be grouped into three broader categories: traditional methods, deep-learning-based methods, and hybrid approaches combining both.

Traditional methods involve predefined (“manual”) features that are often used in conjunction with classic classifiers such as Random Forest and Support Vector Machines (SVM). These methods extract low-level features from video sequences and subsequently make decisions using standardized statistical techniques.

Examples of traditional methods include:

Feature extraction using Histograms of Oriented Gradients (HOG) [9];

Methods for encoding spatio-temporal trajectories [10].

However, traditional methods suffer from difficulties in scalability and insufficient generalization when handling large amounts of heterogeneous data.

Recent research trends highlight the widespread use of deep learning methods, with Long Short-Term Memory (LSTM) networks [11], Convolutional Neural Networks (CNN) [12], and transformers [13] dominating the landscape. These networks excel at automatically learning high-level features directly from raw data, such as video frames, depth maps, or skeletal coordinates.

Deep learning methods deliver superior performance compared to traditional approaches, albeit demanding significant computational resources and requiring large annotated datasets for training.

Hybrid approaches combine the strengths of traditional and deep learning methods. For instance, some methods preprocess data using traditional techniques to extract informative features, which are then passed to a deep neural network for final classification. This allows for merging the interpretability of classical features with the universality of deep learning's representation power.

For example, the algorithm described in [2] for recognizing human-operator actions proceeds as follows: first, a skeleton is constructed based on key points (elbows, wrists, shoulders) using the OpenPose library [14]; then, contextual features are extracted by identifying objects using VGG-16 architecture [15]; finally, classification is performed using a recurrent neural network (RNN) based on an encoder-decoder architecture [16]. Another notable example is the approach presented in [17], which employs a spatial-temporal graph convolutional network (ST-GCN) to analyze upper-body joint positions and arm movements to determine technological operation contexts. Additionally, the method proposed in [18] calculates temporal optical flows (Optical Flow) and processes the data using a multi-layer LSTM network.

Most existing methods confront common challenges, such as illumination changes, interference, occluded body parts, and the necessity of functioning in near-real-time modes. Hence, recognizing human actions remains an ongoing challenge aimed at maximizing accuracy and reliability.

3. Proposed method

In visual sensor-based motion capture systems, human movement data can be obtained by extracting key contextual information from pixels of video or images [1]. Traditional methods of motion perception and recognition heavily rely on manually engineered features, which require prior domain expertise and subjective choices. Deep learning-based approaches overcome these limitations by enabling automatic extraction of meaningful features.

In the original version of the algorithm [19], we extracted motion information using three-dimensional spatio-temporal Gabor filters with fixed parameters. Although these filters possess mathematical interpretability, they have several drawbacks: they are not adaptable to data, not trainable, and their effectiveness strongly depends on precise tuning of hyperparameters (scale, orientation, frequency, etc.).

In the modified version, Gabor filters are replaced with learnable layers of a three-dimensional convolutional neural network (3D-CNN). This approach allows the system to automatically extract spatio-temporal features from video sequences based on the data itself, rather than relying on predefined templates.

The input to the module is a video clip (e.g., consisting of 16 frames, each sized 128×128 pixels), and the output is a feature tensor describing local motion and texture properties.

The proposed modification utilizes a three-dimensional convolutional neural network (3D-CNN) architecture specifically tailored for extracting spatio-temporal features from video sequences. Below is a detailed breakdown of the architecture:

Architecture

- Conv3D Layer #1
- Parameters: kernel size 3×3×3, stride=1, padding=1, filters=32.
- Output: Shape 16×128×128×32.
- Batch Normalization + ReLU Activation
- Normalizes and applies ReLU activation to stabilize and accelerate training.
- Output: 16×128×128×32.
- Conv3D Layer #2
- Parameters: kernel size 3×3×3, stride=1, padding=1, filters doubled to 64.
- Output: Shape 16×128×128×64.
- Batch Normalization + ReLU Activation
- Output: 16×128×128×64.
- MaxPool3D
- Parameters: pool size=2×2×2, reducing each dimension by half.
- Output: 8×64×64×64.

This architecture allows the network to automatically learn and extract the most relevant spatio-temporal patterns, adjusting filters to fit specific actions, scenes, and sensors. Furthermore, these descriptors exhibit versatility and can be trained on diverse data sources, such as RGB, depth, or optical flow.

Using the architecture described above, the 3D-CNN model extracts features from a video sequence belonging to the "push" class from a standard dataset described in reference [20]. The visualization of the model's output features is illustrated in Figure 1.

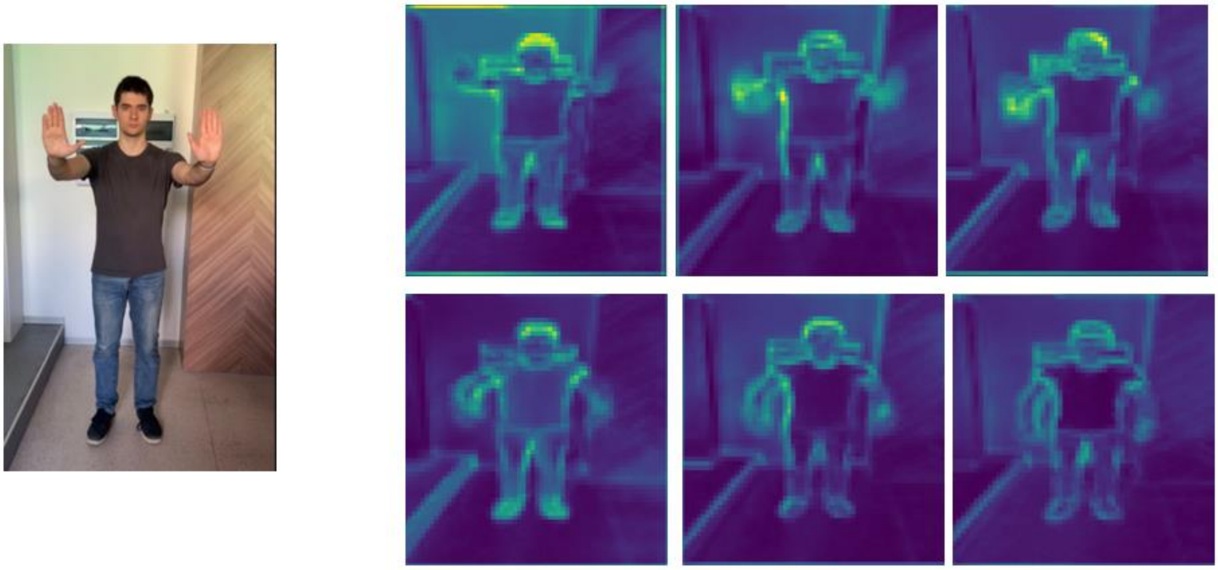

Fig 1. Example of extracting spatio-temporal features using 3D-CNN from a video sequence: the average of the spatio-temporal features over the channels is shown on the right; the left shows 1 frame from the "pushing" video [20]

Here is a detailed explanation.

Let a video clip consist of T frames, each of size H×W, and with either 1 or 3 channels (depending on the modality: RGB or depth). Then:

$$X \in \mathbb{R}^{T\times H\times W\times C}$$

Where X is the input of the first layer. Specifically: T is the number of frames, H and W are the height and width of each frame, C is the number of channels (e.g., 3 for RGB).

Next, a 3D convolution operation is performed, which performs convolution across the spatio-temporal domain.

Let

$$W^{(l)} \in \mathbb{R}^{k_T\times k_H\times k_W\times C_{in}\times C_{out}}$$

Be the filter used at layer l, where: k_T, k_H, k_W are the kernel sizes along the temporal, height, and width dimensions, respectively, $C_{\mathrm{in}}, C_{\mathrm{out}}$are the numbers of input and output channels, respectively.

Given an input tensor

$$X^{(l)} \in \mathbb{R}^{T\times H\times W\times C_{in}}$$

The 3D convolution operation computes as follows:

$$Y^{(l)}{}_{t,h,w,c} == \sum i = 0^{k_T-1} \sum_{j=0}^{k_H-1} \sum_{k=0}^{k_W-1} \sum_{d=1}^{C_{in}} W^{(l)}{}_{i,j,k,d,c} \cdot X^{(l)}{}_{t+i,h+j,w+k,d+b_c}$$

Where: $Y^{(l)}$ is the output of the convolution, b_c is the bias term for channel c.

After the convolution, a nonlinear activation function, such as ReLU, is applied:

$$Z^{(l)} = \sigma(Y^{(l)}) = \max(0, Y^{(l)})$$

Normalization and pooling can then be applied using Batch Normalization across the 5D tensor and 3D MaxPooling:

BatchNorm3D: $\hat{Z}^{(l)} = BN(Z^{(l)})$

MaxPool3D with kernel $p_T \times p_H \times p_W$: $X^{(l+1)}{}_{t,h,w,c} = \max i,j,k\, \hat{Z}^{(l)}_{t+i,h+j,w+k,c}$

The final motion feature tensor is formed after several layers of convolution; we obtain the feature tensor:

$$F\text{motion} = f\text{3DConvNet(X)} \in \mathbb{R}^{T' \times H' \times W' \times D}$$

Where: D — the number of features (tensor depth), $f_{\text{3DConvNet}}$ — the entire function of the 3D network.

The resulting motion feature tensor has the dimensions: $T' \times H' \times W' \times D$. Each element of this tensor describes the dynamics of motion in a local 3D-cube region and contains 64 features (tensor depth) — values that define local spatio-temporal patterns.

This tensor can be used as: local features (for spatial pooling or attention), input to a global classifier, or as one of the streams in a fusion architecture (together with skeleton descriptors, etc.).

The training process is conducted as follows. Let Fmotion proceed to the classifier f_{cls}, which predicts the action class:

$$\hat{y} = f_{\text{cls}}(F_{\text{motion}})$$

The model is trained on a labeled dataset $\{(X_i, y_i)\}i = 1^N$ by minimizing the cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N} \log p\ (y_i|X_i)$$

Where: $p(y_i|X_i) = \text{Softmax}(f\text{cls}(f_{\text{3DConvNet}}(X_i)))$.

Gradients propagate through the entire stack, including the trainable filters $W^{(l)}$, which are updated using gradient descent.

The general block diagram for action classification with the new stage described above is presented in Figure 2.

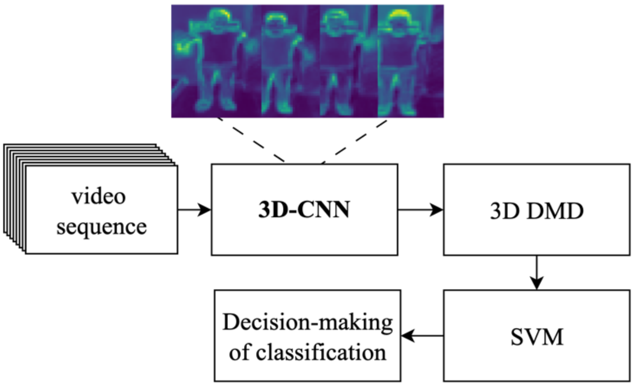

Fig 2. The general block diagram for action classification

The proposed model consists of the following components:

- The input layer accepts video data.
- A trainable 3D convolutional layer replaces the fixed Gabor filter.
- Dense difference of micro-blocks computes a descriptor that describes the actions occurring within the frame [19].
- SVM (Support Vector Machine) is applied for final classification.

Replacing fixed 3D Gabor filters with trainable convolutional filters allows the model to automatically extract optimal motion features, improves adaptation to data, noise, and complex actions, and enhances classification accuracy through deep learning.

To verify the effectiveness of the developed algorithm, a test dataset consisting of 10 classes of gesture commands was collected: "hello," "cross," "ready," "bravo," "push," "swipe left," "swipe right," "fall," "hands up," and "stand." This number of gesture commands is sufficient for implementing interaction with the robot and conducting experimental research. The database was obtained under laboratory conditions using a smartphone camera with specifications of 64 megapixels and an aperture of f/1.8. The object was positioned in front of the camera at a distance of 1.5 to 2 meters. A total of 50 participants performed each action eight times during the preparation of the test set, resulting in 4,000 videos of gesture commands being recorded. Each video does not exceed three seconds in duration. As a result of the experiment, a probability of correct gesture recognition was achieved at 90.3%, which confirms the proper functioning of the developed algorithm under simulated conditions.

SUMMARY AND CONCLUSION

In this paper, an effective approach to human motion recognition in a collaborative human-machine interaction (HRC) system is proposed. The main achievement is the implementation of a three-dimensional convolutional neural network (3D-CNN), which automatically extracts spatio-temporal features from video sequences. The proposed architecture outperforms traditional Gabor filter-based methods.

The combination of trainable convolutional filters and 3D-CNN architecture allows for a significant increase in motion recognition accuracy, reducing the risk of errors and increasing the safety and efficiency of the production process.

Acknowledgment

This work is supported by the Russian Science Foundation under grant No 24-19-00692, https://rscf.ru/project/24-19-00692/

REFERENCES

1.H. Zhou, G. Yang, B. Wang, X. Li, R. Wang, X. Huang, & X. V. Wang, “An attention-based deep learning approach for inertial motion recognition and estimation in human-robot collaboration,” Journal of Manufacturing Systems, 67, 97-110, 2023.

2.J. Hernandez, G. Valarezo, R. Cobos, J.W. Kim, R. Palacios, A.G. Abad, “Hierarchical human action recognition to measure the performance of manual labor,” IEEE Access, 9, 103110-103119, 2021.

3.M.G. Morshed, T. Sultana, A. Alam, Y. K. Lee, “Human action recognition: A taxonomy-based survey, updates, and opportunities,” Sensors, 23(4), 2182, 2023.

4.F. Tassi, F. Iodice, E. De Momi, A. Ajoudani, “Sociable and ergonomic human-robot collaboration through action recognition and augmented hierarchical quadratic programmin,” IEEE/RSJ International Conference on Intelligent Robots and Systems, 10712-10719, 2022.

5.W. Kwon, Y. Jin, S.J. Lee, “Uncertainty-aware knowledge distillation for collision identification of collaborative robots,” Sensors, 21(19), 6674, 2021.

6.G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, P. Steggles, “Towards a better understanding of context and context-awareness,” In International symposium on handheld and ubiquitous computing, pp. 304-307, 1999.

7.P. Wang, H. Liu, L. Wang, R.X. Gao, “Deep learning-based human motion recognition for predictive context-aware human-robot collaboration,” CIRP annals, 67(1), 17-20, 2018

8.H. Liu, L. Wang, “Human motion prediction for human-robot collaboration,” Journal of Manufacturing Systems, 44, 287-294, 2017.

9.T. Surasak, I. Takahiro, C.H. Cheng, C.E. Wang, P.Y. Sheng, “Histogram of oriented gradients for human detection in video,” 5th International conference on business and industrial research (ICBIR), pp. 172-176, 2018.

10.T.V. Nguyen, Z. Song, S. Yan, “STAP: Spatial-temporal attention-aware pooling for action recognition,” IEEE Transactions on Circuits and Systems for Video Technology, 25(1), 77-86, 2014.

11.V. Veeriah, N. Zhuang, G.J. Qi, “Differential recurrent neural networks for action recognition,” IEEE international conference on computer vision, pp. 4041-4049, 2015.

12.S.N. Gowda, M. Rohrbach, L. Sevilla-Lara, “Smart frame selection for action recognition,” AAAI conference on artificial intelligence, vol. 35, No. 2, pp. 1451-1459, 2021.

13.A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, 30, 2017.

14.Z. Cao, G. Hidalgo, T. Simon, S.E. Wei, Y. Sheikh, “Openpose: Realtime multi-person 2d pose estimation using part affinity fields,” IEEE transactions on pattern analysis and machine intelligence, 43(1), 172-186, 2019.

15.K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), pp. 1–1, 2015.

16.K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.

17.M. Delamare, C. Laville, A. Cabani, H. Chafouk, “Graph Convolutional Networks Skeleton-based Action Recognition for Continuous Data Stream: A Sliding Window Approach,” In VISIGRAPP (5: VISAPP), pp. 427-435, 2021.

18.A. Ullah, K. Muhammad, J. Del Ser, S.W. Baik, H.C. de Albuquerque, “Activity recognition using temporal optical flow convolutional features and multilayer LSTM,” IEEE Transactions on Industrial Electronics, 66(12), 9692-9702, 2018.

19.M. Zhdanova, V. Voronin, E. Semenishchev, Y. Ilyukhin, A. Zelensky, “Human activity recognition for efficient human-robot collaboration,” SPIE Artificial Intelligence and Machine Learning in Defense Applications II, vol. 11543, p. 94-104, 2020.

20.A.A. Zelensky, M.M. Zhdanova, V.V. Voronin, E.A. Semenishchev, T.Kh. Abdullin, N.V. Gapon,“Algorithm for recognition of gesture commands based on visual sensors data for contactless robot control systems,” In: Perspective Systems and Control Problems, pp. 9-15, 2023.