Evaluating Input Data Structures for CNN-Based Assembly Action Recognition Using Projected Distance Features

Cosimo Patruno[†1], Mayra Vanessa Alvear Gallón[2,3], Annaclaudia Bono[1,4], Grazia Cicirelli[1]

***Abstract*— Human Action Recognition (HAR) plays a critical role in manufacturing assembly processes, supporting worker safety, operational assistance, production optimization, employee training, and human–robot collaboration. This paper presents a skeleton-based action recognition method that employs a CNN architecture and a distance-based representation of human motion. To encode movements, joint-to-joint distances are first computed from upper-body skeleton data and then projected onto the three Cartesian planes. These features are organized into two different input tensor structures: a one-channel configuration, where projected distances are arranged in a 2D matrix with temporal information explicitly encoded along the column dimension, and a multi-channel configuration, where each channel corresponds to a full frame within a fixed temporal window. The proposed approach is evaluated on the HA4M dataset. The results demonstrate that input tensor structure significantly influences recognition accuracy, and highlight the superior performance of projected distance features compared to traditional distance representations.**

I. INTRODUCTION

Human Action Recognition (HAR) plays a critical role in supporting the transition toward Industry 5.0. In manufacturing assembly processes, HAR contributes to several key objectives, including worker safety and assistance, production optimization, employee training, analysis and improvement of human activities and human–robot collaboration [1], [2], [3]. Among the various industrial processes, assembly tasks are particularly relevant because they form the core of many manufacturing workflows. Developing advanced HAR systems for these tasks is therefore essential to achieve efficient, adaptive, and human-centered production environments. In this scenario, HAR enables robots and intelligent systems to perceive, interpret, and respond to human actions in real time. In manufacturing settings, such capabilities are fundamental not only for monitoring operator activity and ensuring safety, but also for enabling flexible, collaborative robotic behaviors that can adapt to human motion patterns.

Despite its importance, HAR in assembly environments remains highly challenging. Assembly tasks typically involve repetitive, precise motions, small differences between similar actions, and frequent interactions with tools and components. Moreover, variations in workers' execution styles and abilities, along with environmental factors such as lighting changes, occlusions, and cluttered workspaces, further complicate achieving reliable action recognition. Over the past decades, research on HAR for assembly scenarios has expanded considerably, exploring a wide set of sensors and methodologies [2], [4], ranging from vision-based to wearable-sensor-based systems and hybrid solutions, each offering its own strengths and limitations.

Vision-based approaches have been widely used in HAR [5], [6], primarily relying on RGB, RGB-D, or other visual data modalities captured by cameras and depth sensors to interpret and classify human actions. Earlier methods were typically based on hand-crafted features extracted from images, which often required substantial domain expertise and lacked robustness across different environments. In recent years, however, the advancement and diffusion of deep learning have transformed the field. Deep models can automatically learn discriminative features from large volumes of visual data [7], [8], eliminating the need for manual feature engineering. As a result, Deep Neural Networks have become the dominant paradigm for achieving highly accurate recognition of complex actions, particularly in dynamic and industrial settings.

In [9], RGB video streams are used to perform real-time action detection and temporal localization through a deep learning inference module combined with an augmented state-machine mechanism. The system avoids wearable sensors and achieves robust monitoring of complex, fine-grained assembly workflows, demonstrating strong performance in industrial and laboratory datasets. RGB vision data are also used in [8] from the HA4M dataset. The spatiotemporal features are extracted using the I3D network; these features are then structured in different ways to train an MS-TCN++ architecture for assembly action segmentation, highlighting how the organization of RGB-derived visual features can affect recognition performance. In [10], the authors integrate RGB images, depth maps, and 3D skeleton data that are separately processed by deep learning models (CNN-LST and LSTM) to improve the efficiency of gesture recognition in a human–robot collaboration scenario.

Skeleton-based approaches have recently gained significant interest in human action recognition thanks to their robustness to lighting variation, occlusion, and background clutter. In [11], a complete skeleton-centric pipeline is proposed. Specifically, 2D joints of the upper body are first extracted from RGB images, and a 2D-to-3D lifting technique, supplemented with a skeleton data repair mechanism, is then applied to handle missing or corrupted joints. The

[1] Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA) of the National Research Council (CNR), Bari, Italy.

[2] Departamento de Matemáticas y Computación, Universidad de La Rioja, Logroño, Spain

[3] Innovación Riojana de Soluciones IT S.L., Logroño, La Rioja, Spain

[4] Polytechnic University of Bari, Bari, Italy

[†] Corresponding author: cosimo.patruno@cnr.it

obtained joints form a spatial–temporal graph used by a CTR-GCN-Plus network, further enhanced with simplified hand-topology integration to differentiate similar actions. By contrast, in [12], 3D hand-skeleton trajectories are exclusively extracted from 2D RGB streams using MediaPipe Hands. The resulting time series of these fine-grained hand joints are used to classify assembly tasks with an LSTM, emphasizing hand–object interaction as the primary discriminative modality for fine-grained operations. Benmessabih et al. in [13] employ 3D full-body skeleton sequences from industrial datasets, such as InHARD and HA4M, to represent each operator's pose as a graph of joints and bones, exploiting spatio-temporal graph convolution, attention, and multi-resolution temporal modeling to perform frame-level action segmentation in untrimmed industrial videos. The cited studies demonstrate that skeletal data can capture structured motion information in industrial assembly, ranging from detailed hand-joint trajectories to upper-limb or full-body configurations, making it suitable for tasks with varying granularity requirements. Moreover, because skeleton representations skip identifiable visual information, they support privacy and anonymity while enabling accurate characterization of human movements.

This work presents a skeleton-based action recognition approach applied in the challenging context of industrial assembly, where actions are highly similar and subject to operator variability. The method relies on a distance-based motion representation: upper-body joint-to-joint distances are computed and projected onto the three Cartesian planes, producing compact features that enhance the spatial characterization of movement. These features are then organized into two alternative input tensor structures: a one-channel spatio-temporal configuration and a multi-channel projection-based configuration. The main contribution of this work is a systematic investigation of how these different feature organizations impact the performance of CNN architectures, through an extensive evaluation on the HA4M dataset. Experimental results further prove that projected-distance features significantly outperform Euclidean distance features, which fail to fully capture motion variations in 3D space. Overall, the findings highlight the importance of representation design in skeleton-based HAR and offer practical guidance for selecting effective CNN models in assembly-oriented action recognition.

The remainder of the paper is organized as follows. Section II describes the proposed method, detailing the dataset, extracted features, and CNN-based architectures. Section III presents the experimental results, and the discussion of results is given in Section IV. Finally, Section V concludes the paper, focusing on future research directions.

II. METHOD

This section presents the proposed method, giving detailed information on the dataset used to validate our approach, the features extracted, the various configurations of the input data, and the CNN-based architectures employed.

A. Dataset

This study was conducted using the HA4M dataset [14], a multi-modal collection of recordings capturing subjects performing assembly tasks on an industrial object in a laboratory environment. The assembly task comprises a sequence of $N_A = 12$ actions that 41 operators perform at their discretion and preferred, comfortable pace. Each of these actions exhibits varying levels of complexity, resulting in different durations, thus leading to a notable time variance among actions. The data were acquired with a Microsoft Azure Kinect sensor and include RGB images, infrared frames, depth maps, RGB–depth–aligned frames, point clouds, and skeletal representations.

In this work, we focus on the skeleton modality. Skeleton data model the human body as a set of joint coordinates, without maintaining any identifiable visual information [15]. Consequently, skeleton-based human action recognition ensures privacy and anonymity while enabling accurate characterization of human movements.

Fig. 1. Sample frame from the HA4M dataset. The operator stands behind a desk on which all assembly components are arranged. Skeleton joints are superimposed on the operator and numbered sequentially from 1 to 16.

In the HA4M dataset, the skeletal model consists of 32 joints representing the operator's full body. For our purposes, however, we use only $M = 16$ joints, focusing on the upper body, since the task is performed at a desk and the lower body is largely occluded (see Figure 1). This choice is motivated by the nature of the assembly operations, which are mainly executed using the arms, hands, and torso. Consequently, lower-body joints contribute little to action recognition. In addition, reducing the number of joints lowers computational and memory requirements, enabling faster processing.

B. Distance-based Features

In this study, Euclidean distances between pairs of skeletal joints are computed to extract key features that capture the subject's posture during the assembly task. These distances are calculated for each frame in the video, ensuring that joint relationships are consistently preserved over time. For each frame, the total number of distance features is $N_D = \binom{M}{2}$.

Given two distinct skeletal joints j and k, with $j \neq k$, in frame F^i, their 3D coordinates are respectively denoted

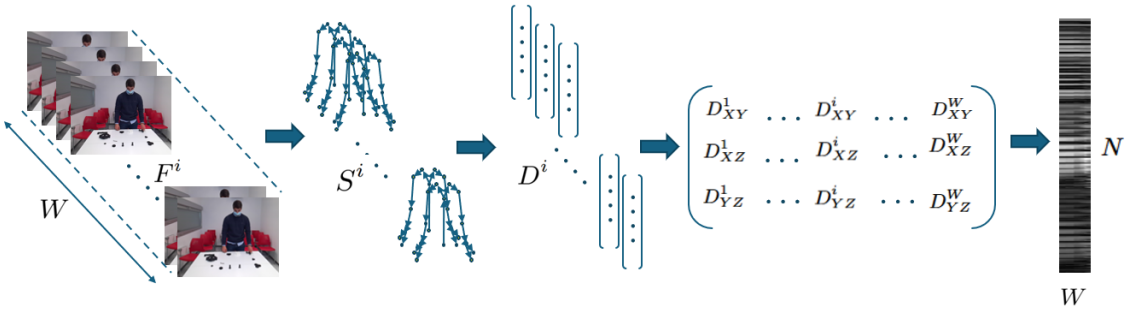

Fig. 2. Input encoding process for the one-channel structure. Starting from the W frames of the sliding window extracted from the video sequence, the 16 upper-body skeleton joints S^i are used to compute the joint-to-joint distances D^i. These distances are then projected onto the three Cartesian planes to obtain the projected distances D^i_{XY}, D^i_{XZ} and D^i_{YZ} for each frame F^i. The resulting tensor has dimensions $W \times N$, where $N = 3N_D$ for the complete set of distances or $N = 3N_R$ for the reduced task-specific subset.

as $J^i_j = (X^i_j, Y^i_j, Z^i_j)$ and $J^i_k = (X^i_k, Y^i_k, Z^i_k)$. Thus, the distance feature $d^i_{j,k}$ can be calculated as follows:

$$d^i_{j,k} = \frac{\|J^i_j - J^i_k\|_2}{d^i_{Norm}} \qquad \forall i \tag{1}$$

where

$$d^i_{Norm} = d^i_{4,11} + d^i_{1,2}$$

denotes the normalization factor introduced to ensure invariance to body size and camera variations. It is defined as the sum of two distances: the distance between the left and right shoulder joints (joints 4 and 11 in Figure 1), and the distance between the pelvis and spine-chest joints (joints 1 and 2 in Figure 1). These distances were chosen because they correspond to rigid torso segments and are less influenced by action dynamics, thereby yielding robust normalization across subjects and camera viewpoints. Given all N_D normalized distances in frame F^i, we denote by D^i the vector containing these distances.

While Euclidean distances between joints provide a compact description of the skeletal pose, they do not fully capture the actual motion of body segments in 3D space. To address this limitation, we consider the projections of joint-to-joint distances onto the three Cartesian planes (XY, XZ, YZ). These projections reflect movement along the three principal directions (frontal (XY), sagittal (XZ), and transverse (YZ)), thus providing a more detailed representation of the subject's motion dynamics. The projected distance vectors derived from D^i, are denoted by D^i_{XY}, D^i_{XZ} and D^i_{YZ}, respectively.

C. Input Tensor Representation

This section outlines the structure of the input tensor provided to the CNN and describes the procedure for extracting relevant data from video frames. For each video sequence, a sliding-window approach is adopted to extract spatio-temporal segments of skeletal data. The sliding window, defined by a length of W frames and a stride of s frames, moves along the temporal axis of the sequence. In our experiments, the window length is set to $W = 15$ covering a time duration of half a second, while the stride is fixed at $s = 1$. The choice of W reflects a compromise between capturing sufficient temporal information to represent actions effectively and selecting the shortest observed action durations across the dataset. The stride s is set to 1 to generate a large number of samples, thereby increasing data richness. Once the W frames are extracted from the video sequence and the corresponding skeletons are obtained, the projected normalized distances D^i_{XY}, D^i_{XZ} and D^i_{YZ} for $i = 1, ..., W$, are used to construct the input data tensor. This tensor is the input to the CNN and can assume two main forms [16]:

1) **One-channel structure:** the tensor is a matrix in which spatial information is encoded along the rows and temporal information along the columns (depth=1);
2) **Multi-channel structure:** the tensor contains spatial information across both rows and columns, while temporal information is encoded along the temporal dimension shared across channels (depth>1).

In both cases, we introduce an additional distinction based on the number of distances considered. One option is to use the full set of normalized distances, $N_D = \binom{M}{2} = \binom{16}{2} = 120$, which accounts for all pairwise distances among the M considered joints. Alternatively, we may focus on a task-specific subset of semantically selected distances. In the assembly task, the actions mainly involve repetitive arm and hand movements. For this reason, we selected a reduced set of N_R distances, corresponding to distances within the arms and between the arms and the torso, yielding $N_R = 70$. Reducing the number of features is advantageous, particularly in applications that require faster computation and lower memory usage.

Table I summarizes the tensor dimensions for both the single-channel and multi-channel structures, considering either the full set of distances or the reduced task-specific subset. Figure 2 and Figure 3 show the complete feature extraction pipeline starting from the video sequence, computing the projected distances between skeleton joints, and finally building the tensor in both the one-channel and multi-channel configurations, respectively.

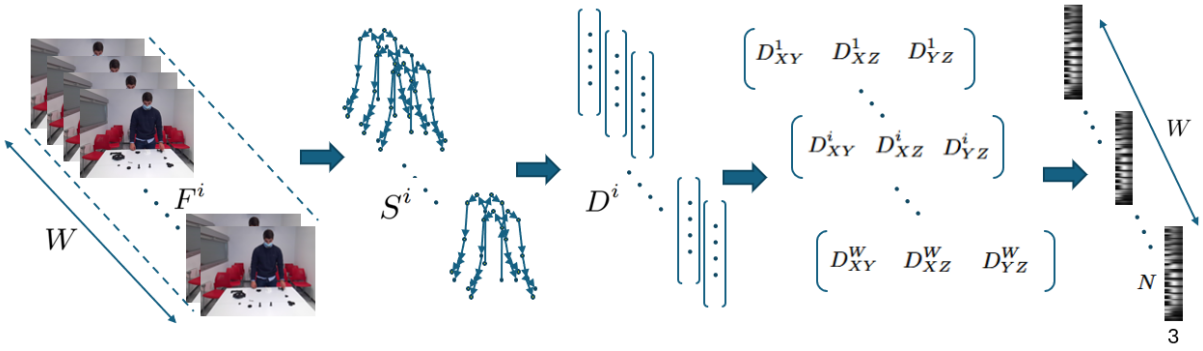

Fig. 3. Input encoding process for the multi-channel structure. Starting from the W frames of the sliding window extracted from the video sequence, the 16 upper-body skeleton joints S^i are used to compute the joint-to-joint distances D^i. These distances are then projected onto the three Cartesian planes to obtain the projected distances D^i_{XY}, D^i_{XZ}, and D^i_{YZ} for each frame F^i. The resulting tensor has dimensions $N \times 3 \times W$, where $N = N_D$ for the complete set of distances or $N = N_R$ for the reduced task-specific subset.

TABLE I

DIMENSION OF THE INPUT TENSOR OF PROJECTED DISTANCES IN THE ONE-CHANNEL AND MULTI-CHANNEL STRUCTURE.

Case	Tensor Dimension
one-channel structure	360×15 $(3N_D \times W)$ 210×15 $(3N_R \times W)$
multi-channel structure	$120 \times 3 \times 15$ $(N_D \times 3 \times W)$ $70 \times 3 \times 15$ $(N_R \times 3 \times W)$

D. CNN Architecture

This section describes the CNN-based neural network architectures used to build the HAR model [16]. Figure 4 presents the proposed network architectures, composed of six layers: the first four are convolutional layers, with the second and fourth each followed by a max-pooling layer. The figure also details the network configuration, specifying the kernel size (Kr), stride size (St), padding size (Pd), and the number of filters (Ft). Each convolutional layer produces a feature map by applying the ReLU activation function. The first and second convolutional layers generate 64 feature maps, while the third and fourth convolutional layers generate 128 feature maps. Max pooling operations downsample feature maps by a factor of 2. The final feature map is then passed through two fully connected layers (FC1 and FC2 in Figure 4), yielding a $1 \times N_A$ score vector, where $N_A = 12$ is the number of actions. This vector is finally converted into a probabilistic action-class prediction via a softmax activation function.

The network architectures, shown in Figure 4, differ in the convolutional layers. In the one-channel configuration (Figure 4(a)), the convolutional filters are 2D kernels that slide vertically and horizontally across the input matrix, learning local patterns that jointly encode spatial relationships among joints and their temporal evolution. In contrast, in the multi-channel configuration (Figure 4(b)), the convolutional filters are 3D kernels designed to process multi-channel inputs,

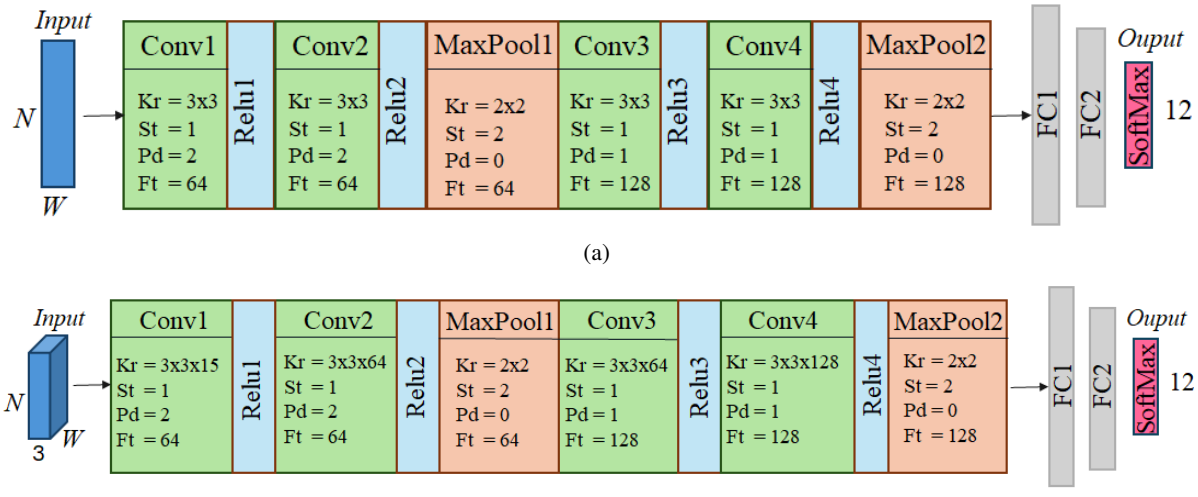

Fig. 4. CNN architectures: (a) *One-channel* case: the input tensor has $N \times W$ dimension (where $N = 3N_D$ or $N = 3N_R$). (b) *Multi-channel* case: the input tensor has $(N \times 3 \times W)$ dimension (where $N = N_D$ or $N = N_R$).

where each channel corresponds to one entire frame within the fixed temporal window. Accordingly, the filters perform convolution only along the spatial dimensions, without sliding over time, thereby preserving the spatial coherence embedded in the complete temporal window. In summary, the fundamental difference between the two architectures lies in the dimensionality of the convolutional filters and the corresponding structure of the input data, which jointly determine how spatial and temporal information is modeled.

III. RESULTS

This section presents the experimental results of the assembly action-recognition task, considering both one-channel and multi-channel input configurations and employing the CNN architectures described in section II-D.

The proposed approach has been evaluated on a subset of the HA4M dataset [14], consisting of 49200 samples selected to ensure a balanced distribution across actions and operators. The data was randomly split into training (60%), validation (10%), and test (30%) sets at sample level. An equal number of samples was generated for each configuration listed in Table I to ensure a fair comparison across settings.

The network architectures were trained on an NVIDIA GeForce RTX 3070 GPU equipped with 8 GB of dedicated memory. Training was performed using stochastic gradient descent with momentum (SGDM). The initial learning rate was set to 0.5 and the momentum coefficient to 0.9. A batch size of 256 was adopted, and the training set was shuffled at every epoch. All experiments were performed using the MATLAB framework.

Training was carried out for a maximum of 500 epochs; however, early stopping was applied by monitoring accuracy and loss on the training and validation sets. Overall, the training accuracy exceeded 98%. The evolution of accuracy across the training and validation sets was closely observed to prevent overfitting. After training, the final model was evaluated on the test set, which was kept completely separate from the training and validation data. In particular, confusion matrices were analyzed for all cases, and key performance metrics were extracted. These metrics include Accuracy, Precision, Recall, and F-score [17].

Table II reports the performance of the CNN for both the one-channel and multi-channel configurations. For completeness, results are provided for feature tensors derived either from Euclidean distances or from projected distances. As shown in the table, the multi-channel configuration consistently achieves higher accuracy, regardless of whether the number of joint-to-joint distances is full (N_D) or reduced (N_R). This confirms that the temporal information encoded in the additional channels of the input tensor (the third tensor dimension) plays a crucial role in improving action-recognition accuracy compared to the one-channel case. Furthermore, the use of projected distances proves more effective, as models trained with projected distance features exhibit superior performance compared to those trained with Euclidean distances.

For completeness, additional experiments were carried out using an LSTM (Long Short-Term Memory) neural network to further investigate the problem. However, recognition performance decreased below 75% in both the one-channel and multi-channel configurations; therefore, these results are not included in the table. Despite LSTMs being specifically designed to capture temporal dependencies in sequential data, they performed poorly in this context. The main limitations may arise from the short observation window of only half a second, which does not provide sufficient temporal depth for effective sequence modeling, and from LSTMs' difficulty in capturing spatial relationships across frames, which are crucial for robust action recognition.

IV. DISCUSSION

The experimental results demonstrate that the choice of input-tensor configuration in a CNN-based deep neural network significantly affects action recognition performance. Furthermore, in the context of assembly action recognition, the multi-channel configuration and the projected distances as features clearly yield better results, with accuracies of 88.35% and 89.69% when using the full (N_D) or reduced

TABLE II

PERFORMANCE METRICS OBTAINED FOR BOTH THE ONE-CHANNEL AND MULTI-CHANNEL INPUT CONFIGURATIONS. GREY ROWS DENOTE THE BEST INPUT CONFIGURATION WITH THE BEST ACCURACIES IN BOLD.

Input Configuration	Model	Features	Precision [%]	Recall [%]	F-score [%]	Accuracy [%]
one-channel						
120×15 ($N_D \times W$)	CNN	Euclidean Distances	80.11	80.08	80.09	80.08
70×15 ($N_R \times W$)	CNN	Euclidean Distances	78.44	78.46	78.45	78.46
360×15 ($3 \cdot N_D \times W$)	CNN	Projected Distances	87.72	87.70	87.71	87.70
210×15 ($3 \cdot N_R \times W$)	CNN	Projected Distances	87.92	87.93	87.93	**87.93**
multi-channel						
$(12 \times 10) \times 15$ ($(N_D) \times W$)	CNN	Euclidean Distances	85.62	85.57	85.59	85.57
$(7 \times 10) \times 15$ ($(N_R) \times W$)	CNN	Euclidean Distances	84.92	84.88	84.90	84.88
$120 \times 3 \times 15$ ($N_D \times 3 \times W$)	CNN	Projected Distances	88.38	88.35	88.37	88.35
$70 \times 3 \times 15$ ($N_R \times 3 \times W$)	CNN	Projected Distances	89.73	89.69	89.71	**89.69**

(N_R) sets of distances, respectively. In contrast, the one-channel configuration achieves 87.70% and 87.93% for the analogous projected distance sets. Overall, projected distances consistently outperform Euclidean distances: in fact, when using Euclidean distances, accuracy does not exceed 80.08% in the one-channel configuration and 85.57% in the multi-channel configuration.

The obtained results highlight several important considerations. In the multi-channel configuration, the network effectively captures meaningful features by exploiting both the temporal evolution of actions, encoded along the channel dimension of the convolutional layers, and the spatial relationships contained within the 2D matrices derived from each frame. In contrast, in the one-channel configuration, the spatial information is represented as a 1D vector, which may limit the network's ability to learn complex spatial dependencies within individual frames. Furthermore, the input configuration impacts the choice of convolutional kernel size. The structure of the input tensor and the kernel configuration are therefore strictly interconnected aspects that determine the consistency of the learned features. Finally, the results also confirm the benefits of projected distances, as they yield better recognition performance than Euclidean distances. This demonstrates that enriching the representation of joint-to-joint relationships via projections provides the network with more discriminative features, thereby enhancing the accuracy of CNN-based action recognition.

V. CONCLUSIONS

This paper presented a comprehensive study on CNN-based action-recognition methods applied to an assembly task. Two major contributions of this work are the investigation of projected distances as input features and the impact of different input-tensor configurations (multi-channel vs. one-channel) on recognition performance. The use of projected distances significantly improved recognition performance as they provide complementary views of the spatial relationships between joints. Despite the short temporal observation window, the experiments confirmed the suitability of CNN architectures for action recognition, with accuracy approaching 90% in the best configurations. Moreover, the analysis demonstrated that the structure of the input data plays a crucial role in determining the effectiveness of CNN-based architectures.

While the proposed methodology has shown strong performance, several promising research directions remain open for exploration. Future studies will investigate alternative deep-learning architectures to better capture complex dependencies among spatial and temporal features. Moreover, extending the observation window is a valuable research direction, as it may enable the detection of a more articulated action dynamics, particularly in the case of longer or complex assembly tasks. Finally, real-time implementation is a fundamental aspect to be explored to enable the deployment of such systems in real industrial environments.

REFERENCES

[1] T. S. Qureshi, M. H. Shahid, A. A. Farhan, and S. Alamri, "A systematic literature review on human activity recognition using smart devices: advances, challenges, and future directions," *Artificial Intelligence Review*, vol. 58, p. 276, 2025.

[2] T. Benmessabih, R. Slama, V. Havard, and D. Baudry, "Online human motion analysis in industrial context: A review," *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107850, 2024.

[3] A. Keshvarparast, D. Battini, O. Battaïa, and A. Pirayesh, "Collaborative robots in manufacturing and assembly systems: literature review and future research agenda," *Journal of Intelligent Manufacturing*, vol. 35, no. 5, p. 2065–2118, 2023.

[4] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human Action Recognition From Various Data Modalities: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3200–3225, 2023.

[5] Y. Zhang and Y. Wang, "A comprehensive survey on RGB-D-based human action recognition: algorithms, datasets, and popular applications," *EURASIP Journal on Image and Video Processing*, vol. 15, 2025.

[6] H. H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Video-based Human Action Recognition using Deep Learning: A Review," 2022. [Online]. Available: https://arxiv.org/abs/2208.03775

[7] L. Romeo, C. Patruno, G. Cicirelli, and T. D'Orazio, "Multi-View Skeleton Analysis for Human Action Segmentation Tasks," in *Proc. of the 14th International Conference on Pattern Recognition Applications and Methods*, 2025, pp. 579–586.

[8] L. Romeo, C. Patruno, G. Cicirelli, and T. D'Orazio, "Deep Learning Methods with Iterative-Boosting for performing Human Action Recognition in Manufacturing Scenarios," in *CoDIT2025 - 11th International Conference on Control, Decision and Information Technologies*, vol. 1, 2025, pp. 835–840.

[9] V. Selvaraj, M. Al-Amin, X. Yu, W. Tao, and S. Min, "Real-time action localization of manual assembly operations using deep learning and augmented inference state machines," *Journal of Manufacturing Systems*, vol. 72, pp. 504–518, 2024.

[10] I. Ding, Jr. and Y.-C. Juang, "A Smart Assembly Line Design Using Human-Robot Collaborations with Operator Gesture Recognition by Decision Fusion of Deep Learning Channels of Three Image Sensing Modalities from RGB-D Devices," *Sensors and Materials*, vol. 36, no. 2, 3, pp. 729–743, 2024.

[11] B. Liu, Y. Yao, H. Wang, Z. He, and A. Dong, "Enhancing industrial human action recognition framework integrating skeleton data acquisition, data repair and optimized graph convolutional networks," *Robotics and Computer-Integrated Manufacturing*, vol. 97, FEB 2026.

[12] O. Ay and E. Emel, "Real-time assembly task validation using deep learning-based object detection and operator's hand-joints trajectory classification," *IEEE Access*, vol. 13, pp. 57 009 – 57 029, 2025.

[13] T. Benmessabih, R. Slama, V. Havard, and D. Baudry, "Graph-based framework for temporal human action recognition and segmentation in industrial context," *Engineering Applications of Artificial Intelligence*, vol. 159, 2025.

[14] G. Cicirelli, R. Marani, L. Romeo, M. G. Domínguez, J. Heras, A. G. Perri, and T. D'Orazio, "The HA4M dataset: Multi-Modal Monitoring of an assembly task for Human Action recognition in Manufacturing," *Scientific Data*, vol. 9, no. 1, p. 745, 2022.

[15] M. V. Alvear Gallón, C. Patruno, G. Mata, C. Domínguez, and G. Cicirelli, "Transformer-based human action recognition for fine-grained industrial assembly tasks," in *IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society*, 2025, pp. 1–6.

[16] C. Patruno, G. Cicirelli, L. Romeo, and T. D'Orazio, "Analysis of Input Data Configurations in CNN-based Human Action Recognition for Assembly Task," in *CoDIT 2025 - 11th International Conference on Control, Decision and Information Technologies*, vol. 1, 2025, pp. 22–27.

[17] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in *Artificial Intelligence Application in Networks and Systems. CSOC 2023. Lecture Notes in Networks and Systems*, vol. 724. Springer International Publishing, 2023, pp. 15–25.