Deep Learning Methods with Iterative-Boosting for performing Human Action Recognition in Manufacturing Scenarios

L. Romeo[†1], C. Patruno[1], G. Cicirelli[1], T. D'Orazio[1]

***Abstract*— The physical and cognitive behavior of humans working in manufacturing environments is an issue that is becoming increasingly important with the advent of Industry 5.0. In this context, the need to monitor operators performing specific tasks is fundamental, particularly when such operators work alongside robots. In fact, guaranteeing the well-being of human workers in industrial scenarios may consistently help in reducing risky and harmful situations. In this work, the HA4M dataset is used to assess human action recognition in a manufacturing environment, where operators perform assembly actions. More specifically, the study has been focused on training a deep learning architecture, namely the MS-TCN++, on RGB data extracted from a specific user, aiming to allow the action recognition model to properly adjust at the selected subject. The RGB data are elaborated using the Inflated 3D model, and the upcoming features have been sorted considering matrix-wise and array-wise dimensions. Furthermore, a 10-stage iterative-boosting technique has been developed, in which the model is iteratively trained by focusing on misclassified samples. It has been proved that the iterative methods allow a faster and more reliable training of the network, reaching an Accuracy, Precision, Recall, and F-score of 70.39%, 74.24%, 68.70%, and 65.73%, respectively, when training using array-wise features. Such results show the effectiveness of the proposed system, laying the foundation for further studies for detecting the operators' actions in the challenging context of Human-Robot Collaboration.**

I. Introduction

Nowadays, Human Action Recognition (HAR) is gaining considerable interest in the literature, as it is a crucial topic in several real-world applications, such as visual surveillance, human-robot interaction, healthcare, and entertainment [1]–[4]. In particular, manufacturing environments may consistently benefit from HAR, especially when Human-Robot Collaboration (HRC) is involved [5].

In recent years, various computer vision solutions for HAR have been implemented, particularly for manufacturing settings [6]. As an example, [7] presents an approach for improving HRC in assembly tasks by using both HAR and object detection. A deep learning model based on graph networks is used to perform action recognition, while a YOLO-based model is used for object detection, aiming to enhance the accuracy and flexibility of HRC in complex assembly settings. Furthermore, [8] presents a system using a Kinect vision sensor and CNN models based on hierarchical clustering to detect different actions in an assembly line. The main goal was to recognize the assembly steps in each station, acknowledge the end of the task, and autonomously notify an Automated Ground Vehicle that must bring the finished product to the designated place. In [9], the authors present a technique to enable robots to recognize and predict human action in assembly tasks. Here, HAR is obtained using deep learning models based on convolutional neural networks (CNN), while Markov Models are used for action prediction. [10] integrates computer vision and deep learning methods by developing an LSTM network combined with CNN architectures. HAR is performed to recognize the actions of operators performing industrial tasks in a collaborative setting, aiming to let the robot execute its task autonomously without requiring explicit human commands.

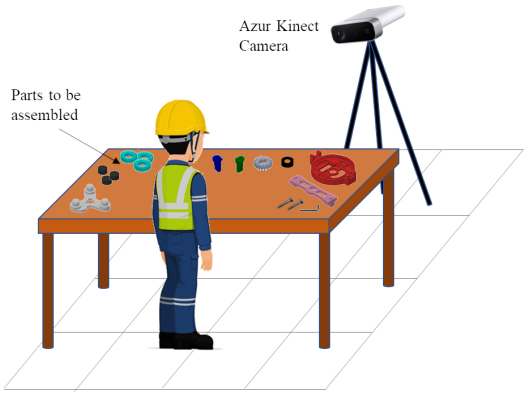

Fig. 1. Representative scheme of the industrial environment depicted in the HA4M dataset [14].

In the literature, there are several studies concerning human action recognition in different fields [11], [12]. Nevertheless, it is still necessary to enlarge the scope of action recognition in manufacturing environments, particularly with the advent of Industry 5.0 in which the human operators represent the core of industry scenarios [13]. This study focuses on training a deep learning model that recognizes the action of a single user within the HA4M dataset [14], aiming to create a system that adapts to the specific movements of each user performing the task, thus increasing the personalization and reliability of the information in HRC contexts. To reach this goal, a Temporal Convolutional Model has been selected, namely the MS-TCN++ [15], and

This research has been partly funded by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by the European Commission under the NextGeneration EU programme.

[1] L. Romeo, C. Patruno, G. Cicirelli, and T. D'Orazio are with the Institute of Intelligent Industrial Systems and Technologies for Advanced Manufacturing (STIIMA) of National Research Council (CNR) of Italy

[†]Corresponding Author: laura.romeo@stiima.cnr.it

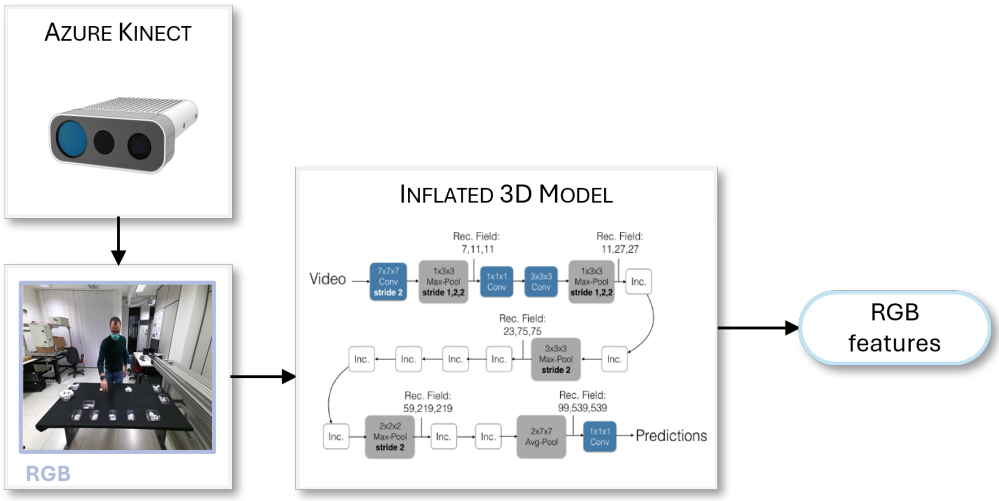

Fig. 2. Depiction of the extraction process performed using the Inflated 3D Model.

specific features have been extracted from the RGB data using the Inflated 3D Model (I3D) [16], aiming to train such architecture using highly discriminant data. A 10-stage iterative-boosting method has been selected for the present study, which is an underexplored topic in the literature on HAR in manufacturing environments [17]. The results have been evaluated considering Accuracy, Precision, Recall, and F-score metrics, comparing the iterative and non-iterative methods. The outcomes highlight how training deep learning models with iterative-boosting methods may lead to high performance by focusing on misclassified samples, while significantly reducing the training time.

The main contribution of the present work are as follows:

- It considers the HA4M dataset [14] to perform HAR, allowing the model to focus on the recognition of the action of a specific user, in a controlled environment.
- It exploits the advantages of the MS-TCN++ architecture for HAR, demonstrating how Temporal Convolutional Networks represent a robust solution for action recognition tasks.
- It proposes a novel approach to manage the features extracted by using the I3D model, structuring them in distinct ways, and sorting them within sliding windows for improved processing.
- It demonstrates the benefits of using an iterative-boosting strategy in HAR models, which enables progressive model refinement, leading to an improved classification performance while reducing training time.

The remainder of the paper is structured as follows. Section II describes the experimental setup developed for the proposed work. First, a depiction of the feature extraction and sorting process is provided. Then, the deep learning architecture is defined, also enlightening the iterative-boosting technique. Furthermore, the metrics considered for the evaluation of the outcomes are presented. Section III reports the experimental results on action recognition, considering both the iterative and non-iterative methods. Finally, Section IV draws the conclusions.

II. Experimental Setup

The proposed work is intended to study, implement, and validate a Human Action Recognition method of a specific assembling task in an industrial production line, within the HA4M dataset [14]. The dataset contains information about operators performing the assembly of an industrial object. An Azure Kinect camera identifies the subject, while the object to be assembled, i.e. an Epicyclic Gear Train, is placed on a working table in front of the operator. Figure 1 represents a scheme of the industrial environment within which the assembly occurs. Each operator follows specific instructions to build the object. The actions are separated into 12 classes plus a "don't care" one. Each frame in each video of the dataset is labeled with a label in the range 0-12. The HA4M dataset contains video information including Skeleton information [18], RGB, Depth, IR, and RGB-Aligned-to-Depth (RGB-A) frames. It must be noted that, for the present work, a single user was taken into account, aiming to create a model that can be adjusted and tuned to recognize the actions of each specific operator.

A. Features Extraction

For the present work, the RGB data have been taken into account, aiming to extract highly discriminant features for training the selected deep learning model. Such types of data are the most used in Action Recognition methods [19], [20], as they allow the extraction and elaboration of highly relevant information. To this aim, as depicted in Figure 2, a set of 1024 features were extracted from the RGB data using the Inflated 3D model [16], which releases highly representative information about the human movements that occur in the RGB frames. Such features have been extracted considering a sliding-window of 15 frames, i.e., 1/2 second. Each window slides within the actions, which have been trimmed singularly from the all videos. The extracted features have been sorted considering two different

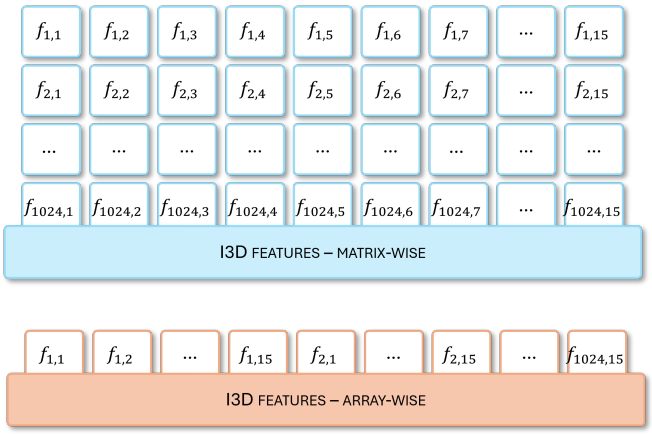

Fig. 3. Representation of the sorting of the features extracted using the I3D model. The first case, in blue, presents the case of a matrix tensor of dimension $[1024 \times 15]$. The second case, in orange, presents the sorting of the array in a 1-dimension tensor of size $[15360 \times 1]$.

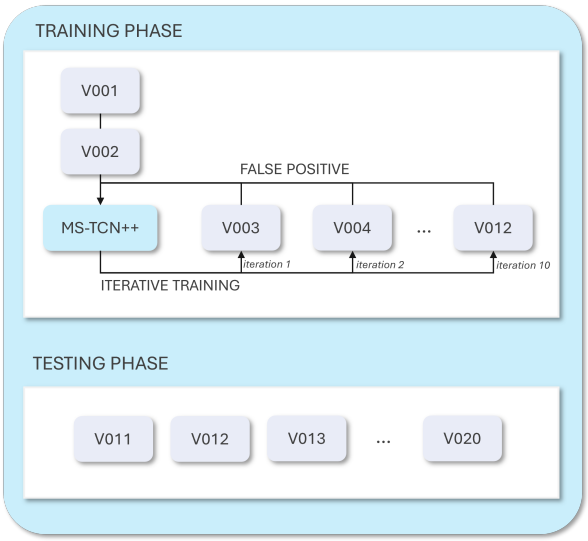

Fig. 4. Pipeline of the training and testing phase.

approaches: matrix-wise and array-wise. In the matrix-wise approach, the features are represented into a 2-dimensional tensor of dimension $[1024 \times 15]$. In the array-wise approach, the features are sorted in a 1-dimensional tensor, thus giving a final dimension of $[15360 \times 1]$. Figure 3 presents a better understanding of such sorting.

B. Deep Learning Model

The user selected for the present work performed the assembly task 20 times within the HA4M dataset. The assembly was performed by mixing the actions across the videos, and while the subject was wearing different types of clothes, aiming to challenge the model and evaluate its capability to generalize. The features extracted from such 20 videos were used to train and test a Temporal Convolutional Network (TCN) based on MS-TCN++ architecture [15]. Such architecture is one of the TCN at the state-of-the-art. TCN-based models are widely used in Action Recognition and Segmentation tasks, as they guarantee high computational performances and thus are useful for real-time applications [21]. Furthermore, the multi-stage structure of the MS-TCN++, together with the dual dilated layers, helps the model to iteratively refine the predictions, increasing the accuracy even when the network is trained with a reduced number of features, but highly representative of human action segmentation tasks. The present work proves how such architecture returns high performances also when used for human action recognition tasks, considering the extracted features to train the MS-TCN++ model, thus obtaining a recognition of the task over a sliding-window of 15 frames.

The model was trained using an iterative boosting technique composed of 10 stages, aiming to enhance the learning efficiency while reducing the computational cost that typically occurs during the training of large datasets. As depicted in Fig. 4, the model was first trained by feeding the features extracted from 2 videos of the selected operator. Then, such model was tested on a set of data among the features extracted from the subsequent 10 videos, considering one video at a time. With this configuration, the initial training set was iteratively increased, adding the wrongly predicted features at each iteration. As for the testing phase, the features extracted from 10 new videos of the same operator were fed into the final deep learning model, obtained after the 10-stage iterations.

C. Evaluation Metrics

To properly evaluate the action recognition performance of the proposed model, four statistical metrics have been considered: Accuracy, Precision, Recall, and F-score. Considering the 12 actions to be recognized, i.e. a multi-class problem, it has been taken into account the evaluation of such metrics as macro-averaged values, aiming to measure how often the model correctly predicts each action among the all instances. With this configuration, the Accuracy can be calculated using the following equation, which allows to understand how well the model correctly classifies the actions within each class.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{FP}_i + \text{TN}_i + \text{FN}_i} \qquad (1)$$

TP_i (True Positive) represents the actions correctly assigned to the class i, FP_i (False Positive) represents the number of actions incorrectly assigned to the class i, FN_i (False Negative) represents the number of actions belonging to the class i, but incorrectly classified, and TP_i (True Negative) represents the number of actions that have not been assigned to class i, nor belong to class i. N represents the total number of actions considered, which in the case of the present study is $N = 12$.

The Precision metric represents the arithmetic mean of the precision scores calculated independently for each class, measuring how often the model correctly predicts each action among the all classes considered. It can be evaluated using

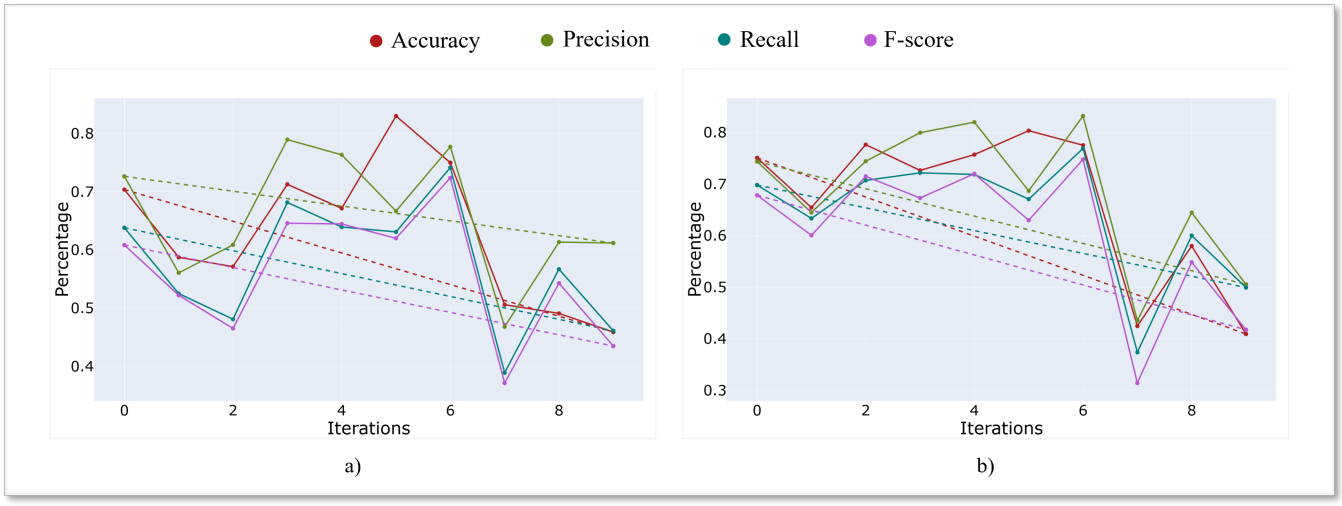

Fig. 5. Results of the 10-stage iterative boosting method, considering the training over the matrix-wise features (a), and the array-wise features (b).

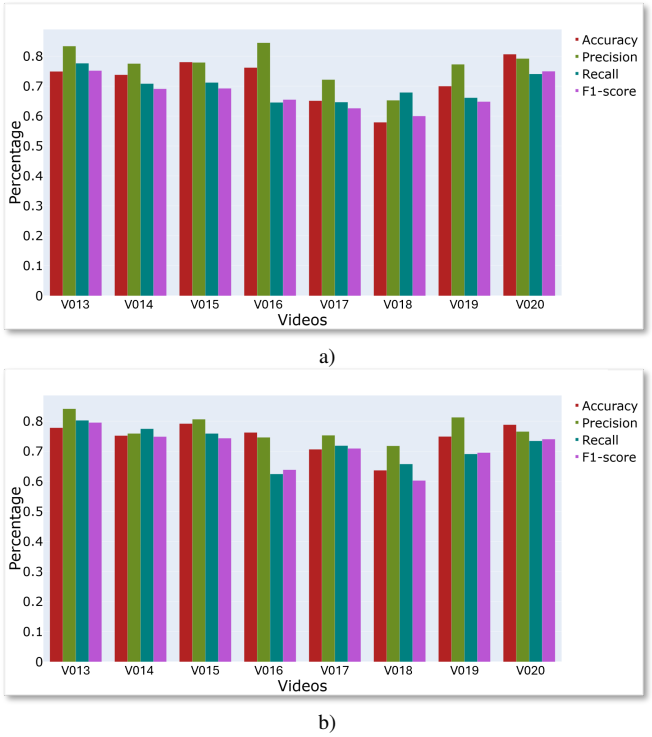

Fig. 6. Results of the testings after training with (a) and without (b) using the iteration boosting method, considering the matrix-wise feature sorting as input data.

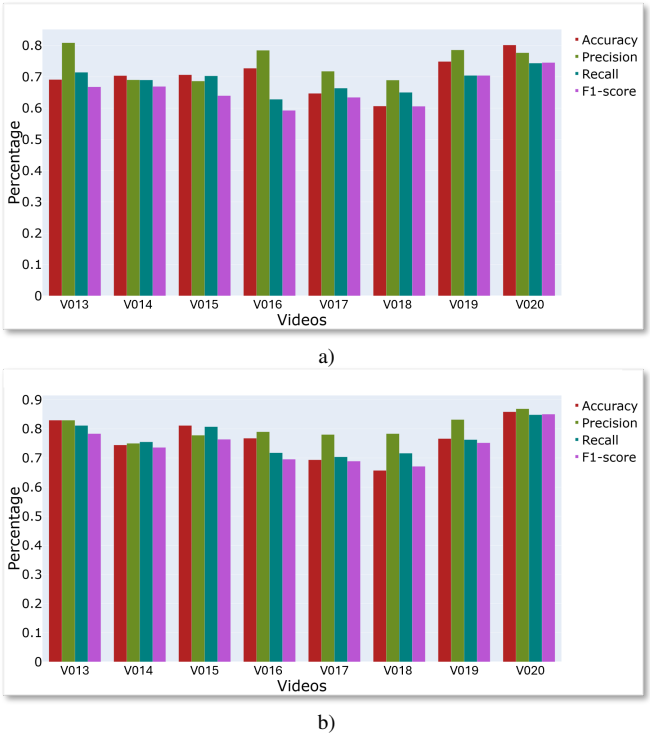

Fig. 7. Results of the testings after training with (a) and without (b) using the iteration boosting method, considering the array-wise feature sorting as input data.

the following equation:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{2}$$

The Recall metric represents the arithmetic mean of the recall scores computed for each class, and evaluates the capability of the model to identify the actions among the classes. It can be evaluated using the following equation:

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{3}$$

Finally, the F-score metric evaluates the balance among Precision and Recall among the all classes. It can be evaluated using the following equation:

$$\text{F-score} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i} \tag{4}$$

III. Results and Discussion

A first analysis of the proposed system was carried out considering the outcomes of each iteration performed. The graphs in Fig. 5 depict the trend of each iteration in terms of Accuracy, Precision, Recall, and F-score. In particular, the graph (a) represents the trend of such metrics considering the model trained over the features sorted in a matrix-wise manner, while graph (b) shows the metrics of the model trained using the array-wise features. Both graphs present a similar trend on all the considered metrics, even though the array-wise features presented slightly higher results. Furthermore, it can be noticed that the 7th iterations suffered a significant drop in both trainings. Such occurrence can be directly linked to the choice to introduce videos in which the subject is wearing a different set and color of clothes. Such change within the features introduced unseen visual data, causing the model to struggle at that specific iteration. Nevetheless, incorporating the wrongly predicted outputs into the training set for the subsequent iterations allows the model to progressively learn to handle heterogenous data.

Fig. 6 and Fig. 7 present the outcomes of the model trained on features with matrix-wise and array-wise sorting, respectively. More specifically, each Figure presents the results obtained from using the 10-stage iterative boosting method (a), and the ones obtained by training the model using directly the features from the entire 12 videos (b). The graphs depict the outcomes of Accuracy, Precision, Recall, and F-score for each video considered during the testing phase. It can be noticed that the model performed heterogeneously among all the videos, proving the ability of the model to properly generalize across different assembly ordering, and different clothing of the subject.

Considering the matrix-wise features sorting, the iterative boosting method reached an average Accuracy of 69.68%, average Precision of 74.67%, average Recall of 66.46%, and average F-score of 64.25%. The model trained without the iteration technique reached 70.52% of average Accuracy, 73.71% of average Precision, 67.19% of average Recall, and 65.69% of average F-score. Taking into account the array-wise sorting of the features, the iterative booster method obtained an average Accuracy of 70.39%, an average Precision of 74.24%, an average Recall of 68.70%, and an average F-score of 65.73%. Finally, the model trained without using the iteration technique reached 76.61% of average Accuracy, 80.15% of average Precision, 76.53% of average Recall, and 74.27% of average F-score.

The presented outcomes enlighten how, in general, the array-wise sorting of the features offers a superior classification accuracy compared to the matrix-wise features. Furthermore, the comparison between the iterative boosting and the not-iterative training underlined that, while iterative boosting offers a more adaptive approach to the refinement of the model, it does not significantly outperform the not-iterative training in terms of accuracy, particularly regarding the array-wise features. Nevertheless, a significative advantage of the iterative boosting method is to progressively refine the model by focusing on misclassified actions, thus improving the decisional process over time. Such technique allows the deep learning model to progressively learn from its mistakes, guaranteeing a better generalization and prediction of the actions to be recognized.

It must be noticed that while the outcomes between the regular model and the iterative booster model are similar, they have been obtained by feeding the latter with a consistently reduced quantity of data, which translates into a significantly reduced training time. The final model of the iterative boosting technique involved 25 hours of training, while the not-iterative model was trained for 64 hours. Such advantage is particularly fundamental when real-time adaptability is required.

IV. Conclusions

The presented study investigated the use of deep learning models with a 10-stage iterative boosting method, aiming to perform HAR in manufacturing scenarios. More specifically, the proposed approach aimed at training the MS-TCN++ architecture to recognize the assembly action of a single user within the HA4M dataset. The model was trained by using discriminant features extracted from RGB data through the I3D model, sorting them in matrix-wise and array-wise formats.

The experimental results proved the efficiency of the proposed iterative boosting method, which allows a progressive model refinement by focusing on misclassified samples. The selected approach was compared with the non-iterative one, proving that the iterative boosting technique provides considerably faster training, which is fundamental in real-time applications. More specifically, the iterative approach required 25 hours of training, while the non-iterative method took 64 hours. Such outcome depicts the importance of iterative methods in manufacturing settings, in which the refinement of deep learning models is crucial, particularly when such information is intended to be sent to a collaborative robot for HRC. In this context, obtaining a refined model through iterative boosting training consistently reduces the training time, guaranteeing high performance.

This work also highlighted the importance of feature representation in HAR, as the matrix-wise and array-wise sorting of the features provided different results. In particular, the array-wise feature sorting showed better classification results, proving to be a more effective format for training the MS-TCN++ architecture for HAR. In the testing phase, such a model achieved 70.39% Accuracy, 74.24% Precision, 68.70% Recall, and 65.73% F-score.

The proposed system presents significant advantages in the field of manufacturing, focusing on HRC. The efficient recognition of the assembly action performed by each operator may facilitate real-time monitoring, enabling adaptive robot responses. Furthermore, the proposed iterative boosting method ensures the continuous learning of the model from operational data, allowing better adaptability to dynamic industrial environments, which is crucial in Industry 5.0 contexts. By enhancing HAR systems in HRC applications,

it is possible to allow machines to understand, predict and support human actions, while increasing productivity and minimizing errors. Further study will explore additional feature analysis for better processing, and the integration of multimodal data to enhance HAR systems in manufacturing settings.

ACKNOWLEDGMENT

The authors are deeply thankful to Michele Attolico and Paola Romano for their technical and administrative support.

REFERENCES

[1] C. Patruno, R. Marani, G. Cicirelli, E. Stella, and T. D'Orazio, "People re-identification using skeleton standard posture and color descriptors from rgb-d data," *Pattern Recognition*, vol. 89, pp. 77–90, 2019.

[2] N. Ma, Z. Wu, Y.-m. Cheung, Y. Guo, Y. Gao, J. Li, and B. Jiang, "A Survey of Human Action Recognition and Posture Prediction," *Tsinghua Science and Technology*, vol. 27, no. 6, pp. 973–1001, 2022.

[3] 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.

[4] 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.

[5] 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.

[6] L. Romeo, R. Marani, A. G. Perri, and J. Gall, "Multi-modal temporal action segmentation for manufacturing scenarios," *Engineering Applications of Artificial Intelligence*, vol. 148, p. 110320, 2025.

[7] Y. Zhang, K. Ding, J. Hui, J. Lv, X. Zhou, and P. Zheng, "Human-object integrated assembly intention recognition for context-aware human-robot collaborative assembly," *Advanced Engineering Informatics*, vol. 54, p. 101792, 2022.

[8] M. Al-Amin, W. Tao, D. Doell, R. Lingard, Z. Yin, M. C. Leu, and R. Qin, "Action recognition in manufacturing assembly using multimodal sensor fusion," *Procedia Manufacturing*, vol. 39, pp. 158–167, 2019.

[9] J. Zhang, P. Wang, and R. X. Gao, "Hybrid machine learning for human action recognition and prediction in assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102184, 2021.

[10] D. Moutinho, L. F. Rocha, C. M. Costa, L. F. Teixeira, and G. Veiga, "Deep learning-based human action recognition to leverage context awareness in collaborative assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 80, p. 102449, 2023.

[11] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1366–1401, 2022.

[12] C. Patruno, V. Renò, G. Cicirelli, and T. D'Orazio, "Multimodal people re-identification using 3d skeleton, depth and color information," *IEEE Access*, 2024.

[13] C. Zhang, Z. Wang, G. Zhou, F. Chang, D. Ma, Y. Jing, W. Cheng, K. Ding, and D. Zhao, "Towards new-generation human-centric smart manufacturing in industry 5.0: A systematic review," *Advanced Engineering Informatics*, vol. 57, p. 102121, 2023.

[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] S.-J. Li, Y. AbuFarha, Y. Liu, M.-M. Cheng, and J. Gall, "MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[16] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[17] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[18] L. Romeo, R. Marani, A. G. Perri, and T. D'Orazio, "Microsoft Azure Kinect Calibration for Three-Dimensional Dense Point Clouds and Reliable Skeletons," *Sensors*, vol. 22, no. 13, p. 4986, 2022.

[19] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, pp. 224–241, Feb. 2011.

[20] B. Filtjens, B. Vanrumste, and P. Slaets, "Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks," *IEEE Transactions on Emerging Topics in Computing*, pp. 1—11, 2022.

[21] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.