

CIRP Conference on Manufacturing Systems (CIRP CMS)

Investigation of continuous human action recognition for manual assembly using the HARDAT dataset

Lukas Büsch^{*,a}, Aaron von Alvensleben^a, Ehssan Roshankar^a, Thorsten Schüppstuhl^a^aHamburg University of Technology, Institute of Aircraft Production Technology, Denickestraße 17, 21073 Hamburg, Germany* Corresponding author. Tel.: +49 40 30601 4241; E-mail address: lukas.buesch@tuhh.de

Abstract

Human Action Recognition (HAR) enables automated systems to interpret human behavior, making it a valuable feedback mechanism for assistance systems in manual assembly. This paper investigates the application of state of the art HAR algorithms, originally developed in domains such as sports, surveillance, and daily activities, to the context of industrial manual assembly. Using the HARDAT dataset, which provides multimodal sensor data from typical assembly tasks, several advanced architectures were evaluated. The focus was placed on four leading models, including graph based and transformer based approaches that process skeletal and RGB input data. Additional established graph based models were included to provide a more comprehensive performance assessment. All models were assessed under continuous recognition conditions using a sliding window approach. The results show that graph based models, particularly ProtoGCN and CTR-GCN, achieved the highest performance in terms of F1 score and accuracy. The findings emphasize the importance of appropriate temporal segmentation and data quality when applying HAR in real world industrial settings. This work also provides reproducible training pipelines to support further research and implementation in assembly assistance systems.

© 2026 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 2026 CIRP CMS

Keywords: Human Action Recognition; Assembly; Continuous Action Recognition; Methods-Time-Measurement; Assistance System; Artificial Intelligence

1. Introduction

Effective human-machine collaboration in industrial assembly requires assistance systems that can adapt to dynamic environments and respond intelligently to human actions. Human Action Recognition (HAR) is a key enabling technology, allowing such systems to observe, interpret, and react to worker activities in real time [38]. Accurate HAR supports context-aware assistance during manual and semi-automated tasks, improving flexibility, safety, and overall process efficiency.

HAR relies on classifying human actions from sensor data [6], with its performance depending on robust models and high-quality training datasets. The Methods-Time Measurement (MTM) framework [22] provides a structured approach by segmenting motions into time-standardized elements, facilitating modeling and interpretation of worker activities. However, suitable training data remains limited: existing assembly datasets are often limited in scope or rely on pre-segmented clips, restricting the generalization of trained models to real-world applications, and continuous recognition in unsegmented streams remains largely unaddressed [7]. While deep learning

models perform well on segmented tasks, their robustness under online conditions is less established. The objective of this work is to determine the extent to which existing models can perform on such data and to analyze the limitations that impede further progress. Therefore, a comparative evaluation of state-of-the-art HAR models for recognizing continuous assembly-related human actions was conducted. Special attention was given to hyperparameter tuning to optimize performance for the specific challenges of industrial assembly motions.

2. Related Work

State-of-the-art HAR systems rely on models that capture spatio-temporal patterns and on large annotated datasets. While HAR has advanced in domains like sports, daily activities, and surveillance [35, 45, 19], its application in industrial assembly remains challenging due to limited access to suitable data. Industrial datasets must include multimodal sensor data, varied conditions, real-world variability, and reproducible setups [7], yet data collection is constrained by privacy concerns, operational disruptions, and labeling difficulties [4].

2.1. Availability of Training Data

Recent industrial HAR datasets [34, 10, 1, 32, 5, 43, 11] offer multimodal recordings, fine-grained annotations, varied viewpoints, hand–object interactions, and human–robot collaboration scenarios. Despite these advances, limitations remain: class imbalance and the focus on generic rather than assembly-specific actions challenge model performance, while non-standardized labeling hinders cross-dataset training and transfer learning [7]. Moreover, much existing HAR research focuses on segmented recognition using pre-segmented clips, which does not reflect real-world industrial workflows. In practice, continuous recognition is required, where actions must be identified in real time from unsegmented streams with ambiguous boundaries and overlapping motions [28, 12].

The HARDAT dataset addresses these issues by providing a more balanced and standardized benchmark for manual assembly HAR [7]. It includes 14 participants performing generic assembly tasks lists, including tasks such as *measuring*, *hammering*, and *drilling*. Their fine-grained actions are mapped to six basic MTM-1 movements [22], such as *Move* or *Position*, with an additional *None* class for undefined actions. Data was collected at 30 FPS using Microsoft Azure Kinect, with joint coordinates extracted via Microsoft’s Body Tracking SDK [29]. The 99.7 GB dataset includes 70 sequences (11.78 hours) of synchronized 3D skeletal graphs, 420p depth, and 720p RGB video. Its standardized, assembly-specific focus and expandability motivates its use in this work.

2.2. Continuous Human Action Recognition

Continuous HAR involves recognizing human actions in unsegmented data streams without predefined boundaries, a task complicated by smooth, ambiguous, or person-dependent transitions [28, 12, 13]. Performing continuous HAR on a (near-)real-time data stream is commonly referred to as online recognition. Most existing research focuses on segmented HAR, training models on isolated, pre-cut clips containing a single labeled action, which fails to capture the overlap, transitions, and ambiguity of real-world motion, as in datasets like NTU RGB+D [35, 26]. In industrial settings, however, continuous recognition is required to support real-time monitoring and interaction in tasks such as assembly or inspection [28].

Most models require fixed-length inputs, posing a challenge for continuous HAR. Therefore, the Sliding Window (SW) approach is commonly used to segment streams into model-compatible clips [12, 13] (see Fig. 1). SWs are labeled via majority vote, where the stride controls overlap: smaller strides improve temporal coverage, while larger strides reduce computational cost. Since SW size strongly affects HAR performance, careful tuning is essential for effective recognition.

2.3. HAR Network Architectures

For RGB+D and skeleton data, CNNs [17, 41] extract spatial-temporal features, while GCNs [25, 36, 27, 8, 42, 37] leverage pose graph structures, and transformers [24, 15] cap-

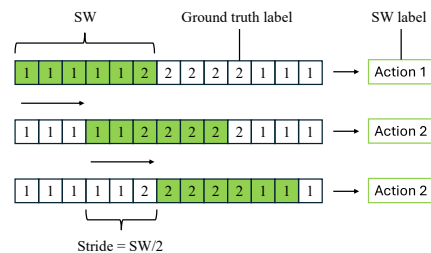


Fig. 1: Sliding window method with majority voting for data segmentation.

ture long-range dependencies in spatial and temporal dimensions. In skeleton-based HAR, ST-GCN [42] introduced spatial-temporal graph convolutions, extended by multi-stream models like 2s-AGCN [36] with joint (J), bone (B), and motion-enhanced (JM, BM) streams[37]. ProtoGCN [25] further adds K-Bone (K) and K-Bone Motion (KM) [9] streams to model relative joint positions and dynamics. MS-G3D [27] introduces multi-scale temporal graphs, and CTR-GCN [8] refines graph topology for improved performance on large datasets. Transformer-based models such as Skeleton MixFormer [40] and Hyperformer [44] enhance spatial-temporal modeling via mix-attention or hypergraph self-attention, while JMDA [39] adds joint motion difference augmentation for robustness. RGB-based approaches like RGBPoseC3D [17] and π -ViT [33] combine RGB and pose data to refine spatiotemporal attention, achieving strong results on benchmarks such as NTU RGB+D [35, 26].

The preceding sections highlighted core challenges in applying HAR to industrial assembly: limited standardized real-world data and the need for online recognition. While many methods exist for other domains, their use in continuous assembly scenarios is underexplored. [28] compared several GCN architectures on assembly-specific HA4M for skeleton-based continuous action recognition, showing the potential of pose-based models but focusing solely on GCNs.

This paper evaluates four state-of-the-art architectures, two skeletal-based and two RGB-based, on continuous data streams. The models were selected based on performance rankings on the NTU RGB+D 120 dataset [31, 30], representing a diversity of spatial and temporal modeling approaches. DSCNet and other top-tier models were excluded due to lack of publicly available code or excessive VRAM requirements that exceeded our hardware capacity. Since the RGBPoseC3D model is implemented within the PySkl framework [16], which also includes established GCN models such as ST-GCN and MSG3D, these were included as representative baselines. An overview of the networks compared in this paper is shown in Table 1.

3. HAR Model Evaluation Framework

This section presents the methodological framework, which centers on two key components: the analysis and preparation of the input data, and the systematic tuning of hyperparameters to enable a reliable comparison of model performance.

Table 1: HAR networks compared in this paper.

Architecture	Used Data Modalities
ProtoGCN [25]	Pose Graph
JMDA [39]	Pose Graph
RGBPoseC3D [17]	RGB + Pose Graph
π -ViT [33]	RGB + Pose Graph
MS-G3D [27]	Pose Graph
CTR-GCN [8]	Pose Graph
ST-GCN++ [16]	Pose Graph
MS-AGCN [37]	Pose Graph

3.1. Data Analysis and Preparation

A preliminary analysis of the HARDAT dataset was conducted to guide SW selection and assess class imbalance. As shown in Fig. 2, the actions *Grasp* and *Release* are underrepresented due to their short duration, making it challenging to capture them within typical window lengths without including irrelevant context. The remaining action classes have longer median durations, allowing for potentially more stable representation within each sample segment.

The remaining actions have longer median durations, supporting more stable segment representations, while the infrequent None class, despite its longer duration, is expected to have limited impact on training. Across all classes, median and average action durations are 90 and 102 frames, respectively, providing a reference for defining suitable window sizes [7].

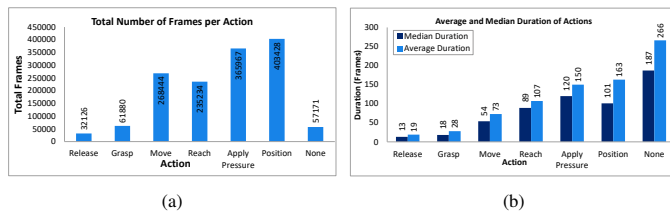


Fig. 2: Total amount of frames (a), median and average duration (b) for each action class in the HARDAT Dataset.

To prepare the dataset, all modalities were aligned frame-wise. Skeletal data and labels were stacked into NumPy arrays, and sliding windows were applied to the RGB and skeleton sequences with labels assigned by majority voting. Ties were resolved by the label of the chronologically first frame. A fixed 55/15 training/validation split was used. As suggested by [7], 26 of 32 skeletal joints were used, excluding the lower body.

Data clips were then converted into the input formats required by each network and annotated for compatibility with the respective training pipelines [2, 20, 14, 18]. For PySkl-based GCNs, temporal augmentation was replaced with a custom function to preserve the SW structure and resolution to prevent unintended resampling. Finally, all training pipelines were adapted to HARDAT, including data loading, preprocessing, graph alignment, and output class configuration.

3.2. Hyperparameter Optimization

The hyperparameter optimization was performed using Optuna [3] to identify optimal training conditions for each architecture. The search targeted both temporal slicing (SW size and stride) and training parameters (batch size, optimizer, learning rate), which strongly influence HAR performance [13, 28, 23]. A preliminary screening revealed that parameters such as momentum and weight decay had little impact and were therefore excluded. The optimization space is summarized in Table 2.

Table 2: Hyperparameters and their respective search space for optimization.

Hyperparameter	Optimization Space
Sliding Window Size	[60, 90, 120]
Stride	$[\frac{SW}{2}, \frac{SW}{3}]$
Batch Size	[8, 16, 32]
Optimizer	[SGD, Adam]
Learning Rate (SGD)	$[10^{-3} \text{ to } 10^{-1}]$
Learning Rate (Adam)	$[10^{-5} \text{ to } 10^{-3}]$

Model performance was primarily evaluated using Top-1 accuracy, with F1-score tracked to account for class imbalance and evaluate performance on underrepresented or short-duration action classes.

A key preprocessing variable is the sliding-window size. With HARDAT recorded at 30 FPS and a median action duration of 90 frames, window sizes of 60, 90, and 120 frames were tested, following prior work suggesting that windows aligned with the median action duration perform best [12]. Shorter windows could benefit brief actions such as *Grasp* and *Release*, but preliminary tests showed they substantially increased errors on longer actions, so window sizes below 60 frames were excluded. To balance temporal resolution and runtime, the stride was set to half or one-third of the window size. While half-window strides are common and computationally efficient, the smaller one-third stride was evaluated to test whether denser sampling improves performance, despite its higher training cost.

Additional tunable parameters in the training loop were batch size, learning rate, and optimizer. Batch sizes of 8, 16, 32 were tested, following common practice and original implementation settings. Larger batches were excluded due to computational cost. Two optimizers were considered: SGD and Adam. Learning rates for SGD were sampled from 10^{-3} to 10^{-1} , and for Adam from 10^{-5} to 10^{-3} , reflecting their respective sensitivity and convergence characteristics [21]. Model-specific epoch counts were increased until signs of overfitting appeared.

The search space was limited by available computational resources (Section 4). High-cost RGB branches (π -ViT and RGBPoseC3D) were excluded, and optimization focused on the graph-based models, with MS-G3D serving as a representative for the older GCN architectures in PySkl.

4. Experiments and Discussion

This section outlines the hardware used for training and evaluation and summarizes the results of the hyperparameter optimization and subsequent model comparisons. Experiments were run on two systems: a workstation (Intel Core i9-12900K, NVIDIA RTX A5000 (24 GB), 64 GB RAM) and a cluster node (AMD EPYC Milan, 3 × NVIDIA RTX 4090 (24 GB each), 78 GB RAM). ProtoGCN and JMDA were trained on the workstation using a single GPU, while all PySkl-based architectures and π -ViT were trained on the cluster node with distributed multi-GPU setups, required especially for RGB-based models due to their higher complexity, additional modalities, and substantially greater memory demands.

4.1. Hyperparameter Optimization Results

The best-performing hyperparameter configurations in terms of Top-1 accuracy during validation for ProtoGCN, JMDA, and MSG3D are summarized in Table 3. All three models converged on the same window size (120 frames) and stride ($\frac{SW}{3}$), indicating a shared sensitivity to temporal segmentation. However, batch size, optimizer, and learning rate differed across models. The hyperparameter search revealed that temporal segmentation was the dominant factor across all models. Window size had the strongest influence on performance, with relative importance values of 0.55 for ProtoGCN, 0.81 for JMDA, and 0.42 for MSG3D, followed by stride (0.21 / 0.11 / 0.31). In contrast, training-specific parameters like batch size (0.10 / 0.05 / 0.20) and optimizer and learning rate (0.14 / 0.03 / 0.07) had only minor effects. Based on these findings, the optimized configurations were used for subsequent training: the MSG3D setup was transferred to the other PySkl-based GCNs, while the RGB+Pose models (π -ViT and RGBPoseC3D) followed their original training schemes, using SW=90 and a stride of $\frac{SW}{2}$ due to GPU memory constraints.

Table 3: Hyperparameter optimization results.

Hyperparameter	ProtoGCN	JMDA	MSG3D
Window Size	120	120	120
Stride	$\frac{SW}{3}$	$\frac{SW}{3}$	$\frac{SW}{3}$
Batch Size	16	32	8
Optimizer	SGD	SGD	Adam
Learning Rate	0.021	0.079	0.00043

4.2. Classification Results and Model Comparison

Table 4 summarizes the best Top-1 accuracy and F1-scores for all models. CTR-GCN achieved the highest accuracy (0.691), closely followed by MS-AGCN, ProtoGCN and ST-GCN++. ProtoGCN obtained the best F1-score (0.4752). Although accuracies lie in a narrow range, the results indicate that graph-based models outperform transformer-based and RGB-enhanced architectures on HARDAT. The consistently low F1-scores highlight prediction imbalances, likely caused by short

or underrepresented classes such as *Grasp* and *Release* which challenge the models' ability to maintain high recall and precision simultaneously. Since the best GCN results rely on multi-

Table 4: Best classification result for each model.

Model	Top-1 accuracy	F1-score
ProtoGCN	0.6908	0.4752
JMDA	0.6759	0.2866
RGBPoseC3D	0.6538	0.4081
π -ViT	0.5874	0.3869
MS-G3D	0.6701	0.436
CTR-GCN	0.6910	0.4419
ST-GCN++	0.6866	0.4425
MS-AGCN	0.6909	0.4531

stream fusion (J, B, JM, BM, and for ProtoGCN also K and KM), Table 5 further breaks down stream-wise performance: ProtoGCN yields the strongest single-stream J-performance, while CTR-GCN benefits most from multi-stream fusion.

Table 5: Per-stream accuracy and F1-score for the two best performing GCNs.

Model	Stream	Top-1 accuracy	F1-score
ProtoGCN	Joint	0.6776	0.4702
	Bone	0.6512	0.4401
	K-Bone	0.6646	0.4626
	Bone-Motion	0.6387	0.4203
	Joint-Motion	0.6232	0.4320
	K-Bone-Motion	0.6332	0.4319
	Fusion	0.6908	0.4752
CTR-GCN	Joint	0.6555	0.4374
	Bone	0.6443	0.4073
	Joint-Motion	0.6470	0.4158
	Bone-Motion	0.6210	0.3882
	Fusion	0.6910	0.4419

4.3. Discussion

The classification results in Table 4 show that all models achieve accuracies within a narrow range, but F1-scores differ markedly. High accuracy is primarily driven by frequent, long-duration actions (e.g., *Reach*, *Position*, *Apply Pressure*, with 75–83% accuracy) while the significantly lower F1-scores expose poor performance on critical, short-duration actions such as *Grasp* and *Release*, which are frequently misclassified (see Fig. 3). These imbalances stem from both the short duration and similarity of certain actions and from the imbalanced class distribution in the dataset. Since sliding-window labels are assigned by majority voting, the underlying class imbalance (see Fig. 2) directly propagates into training, making high recall and precision for underrepresented actions difficult and resulting in consistently low F1-scores.

The SW approach itself introduces further limitations. The window size, as one of the most influential hyperparameters,

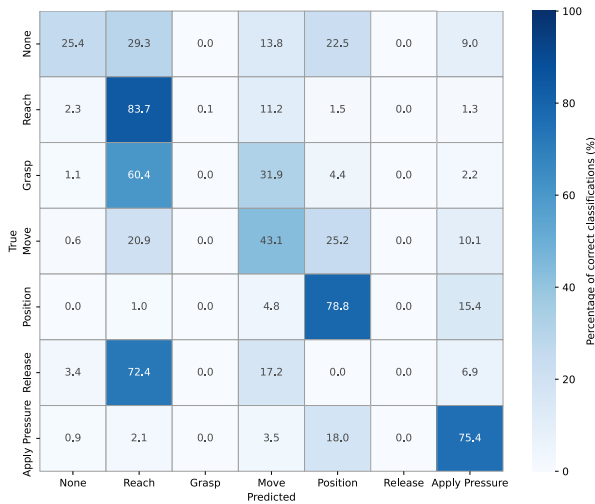


Fig. 3: Confusion matrix for the CTR-GCN classification results.

must balance sensitivity to short actions against stable recognition of longer ones. Larger windows favor temporal consistency in long-duration actions, which inflates accuracy, but they suppress F1-scores by filtering out brief, transient motions through majority voting. Smaller strides increase temporal resolution but sharply raise computational cost. Thus, while sliding windows remain necessary for current online HAR approaches, they are a suboptimal solution for datasets with highly variable action durations.

Additional factors also impacted performance. Some HAR-DAT action labels were inconsistently annotated or corrupted, which likely degraded training quality. Hardware limitations constrained π -ViT and RGBPoseC3D to smaller windows and larger strides, disadvantaging them relative to more memory-efficient GCNs like ProtoGCN and CTR-GCN. Larger sliding windows consistently improved GCN accuracy, indicating that longer temporal context is beneficial and would likely also enhance RGB-based models, though this could not be tested. The strong performance of GCNs despite moderate dataset size suggests that their inductive biases toward structured spatial data align well with pose data, whereas transformer-based models may require larger, more diverse datasets to develop their full potential.

Overall, the narrow accuracy range across all models indicates that architectural choices alone are unlikely to yield significant gains on HARDAT. Meaningful improvements will instead depend on addressing dataset-related limitations such as action-duration imbalance, inconsistent labels, and limited scale. While a 0.691 Top-1 accuracy provides a baseline for process monitoring, real-time assistance requires higher precision to maintain user trust and avoid disruptive false prompts. The 0.4752 F1-score highlights the challenge of detecting brief or rare assembly steps. This difficulty is common in industrial benchmarks and marks a clear threshold for future data-centric optimization [11].

5. Conclusion and Outlook

This paper compared state-of-the-art HAR architectures on the HARDAT dataset to assess their suitability for industrial assembly tasks. Among the evaluated models, the graph-based CTR-GCN and ProtoGCN achieved the highest accuracy and F1-score, respectively. However, top-performing methods showed only slight differences in Top-1 accuracy and appear to approach a performance bound. Therefore, improvements are expected from dataset enhancements, such as increased scale, balanced classes, and accurate labeling, rather than architectural changes. Statistical significance testing is recommended for further model evaluation. This work provides reproducible pipelines, supporting future benchmarking and practical adaptation. Future research should explore alternatives to fixed sliding windows, such as LSTM-based approaches, and consider computational efficiency for online deployment while leveraging greater resources for high-capacity models.

Acknowledgements

This work is co-funded by the Free and Hanseatic City of Hamburg and the European Regional Development Fund (ERDF) under the InnovationFocus Green Aviation Technologies (GATE II) program.



References

- [1] Aganian, D., Stephan, B., Eisenbach, M., Stretz, C., Gross, H.M., 2023. Attach dataset: Annotated two-handed assembly actions for human action understanding. [arXiv:2304.08210](https://arxiv.org/abs/2304.08210).
- [2] aidarikako, 2024. JMDA: Joint mixing data augmentation for skeleton-based action recognition (github repository). <https://github.com/aidarikako/JMDA>. GitHub repository, accessed 11/06/25.
- [3] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery. URL: <https://doi.org/10.1145/3292500.3330701>, doi:10.1145/3292500.3330701.
- [4] Al-Amin, M., Qin, R., Moniruzzaman, M., Yin, Z., Tao, W., Leu, M.C., 2021. An individualized system of skeletal data-based cnn classifiers for action recognition in manufacturing assembly. *Journal of Intelligent Manufacturing* doi:10.1007/s10845-021-01815-x.
- [5] Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., Gould, S., 2020. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. *International Journal of Computer Vision* URL: <https://ikeaasm.github.io>.
- [6] Büsch, L., Koch, J., Schoepflin, D., Schulze, M., Schüppstuhl, T., 2023. Towards recognition of human actions in collaborative tasks with robots: Extending action recognition with tool recognition methods. *Sensors* 23. doi:10.3390/s23125718.
- [7] Büsch, L., Palazoglu, M., Schüppstuhl, T., 2024. HARDAT: Human action-recognition dataset for manual assembly tasks, in: Proceedings of the Global Conference on Sustainable Manufacturing (GCSM 2024).
- [8] Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W., 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). URL: <https://github.com/Uason-Chen/CTR-GCN>.
- [9] Chi, H., Ha, M.H., Chi, S., Lee, S.W., Huang, Q., Ramani, K., 2022. Infogcn: Representation learning for human skeleton-based action recogni-

- tion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:[10.1109/CVPR52688.2022.01955](https://doi.org/10.1109/CVPR52688.2022.01955).
- [10] Cicirelli, G., Marani, R., Romeo, L., Domínguez, M.G., Heras, J., Perri, A.G., D'Orazio, T., 2022. The ha4m dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing. Scientific Data doi:[10.1038/s41597-022-01843-z](https://doi.org/10.1038/s41597-022-01843-z).
- [11] Dallel, M., Havard, V., Baudry, D., Savatier, X., 2020. Inhard: Industrial human action recognition dataset in the context of industrial collaborative robotics, in: Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE). URL: <https://recherche.cesi.fr/inhard-industrial-human-action-recognition-dataset/>.
- [12] Dallel, M., Havard, V., Dupuis, Y., Baudry, D., 2022. A sliding window based approach with majority voting for online human action recognition using spatial temporal graph convolutional neural networks, in: Proceedings of the 7th International Conference on Machine Learning Technologies (ICMLT 2022), ACM. doi:[10.1145/3529399.3529425](https://doi.org/10.1145/3529399.3529425).
- [13] Delamare, M., Laville, C., Cabani, A., Chafouk, H., 2021. Graph convolutional networks skeleton-based action recognition for continuous data stream: A sliding window approach, in: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) – Volume 5: VISAPP, SCITEPRESS. doi:[10.5220/0010234904270435](https://doi.org/10.5220/0010234904270435).
- [14] dominickrei, 2024. pi-vit: Pose induced video transformers for understanding activities of daily living (github repository). <https://github.com/dominickrei/pi-vit>. GitHub repository, accessed 11/06/25.
- [15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlby, N., 2021. An image is worth 16×16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations (ICLR). Published as a conference paper at ICLR 2021.
- [16] Duan, H., Wang, J., Chen, K., Lin, D., 2022a. Pyskl: Towards good practices for skeleton action recognition. [arXiv:2205.09443](https://arxiv.org/abs/2205.09443).
- [17] Duan, H., Zhao, Y., Chen, K., Lin, D., Dai, B., 2022b. Revisiting skeleton-based action recognition. [arXiv:2104.13586](https://arxiv.org/abs/2104.13586).
- [18] firework8, 2024. ProtoGCN: A prototypical perspective for skeleton-based action recognition (github repository). <https://github.com/firework8/ProtoGCN>. GitHub repository, accessed 11/06/25.
- [19] Jegham, I., Khalifa, A.B., Alouani, I., Mahjoub, M.A., 2020. Vision-based human action recognition: an overview and real world challenges. Forensic Science International: Digital Investigation doi:[10.1016/j.fsidi.2019.300889](https://doi.org/10.1016/j.fsidi.2019.300889).
- [20] kennymckormick, 2022. pyskl: A python library for skeleton action recognition (github repository). <https://github.com/kennymckormick/pyskl>. GitHub repository, accessed 11/06/25.
- [21] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [22] Koch, J., Büsch, L., Gomse, M., Schuppstuhl, T., 2022. A methods-time-measurement based approach to enable action recognition for multi-variant assembly in human-robot collaboration, in: Procedia CIRP, 106th Conference on Assembly Technology and Systems, Elsevier. doi:[10.1016/j.procir.2022.02.184](https://doi.org/10.1016/j.procir.2022.02.184).
- [23] Koh, T.C., Yeo, C.K., Jing, X., Sivasdas, S., et al., 2023. Towards efficient video-based action recognition: Context-aware memory attention network. SN Applied Sciences 5.
- [24] Leite, C.S., Mauranen, H., Zhanabatyrova, A., Xiao, Y., 2024. Transformer-based approaches for sensor-based human activity recognition: Opportunities and challenges.
- [25] Liu, H., Liu, Y., Ren, M., Wang, H., Wang, Y., Sun, Z., 2024. Revealing key details to see differences: A novel prototypical perspective for skeleton-based action recognition.
- [26] Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C., 2019. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE transactions on pattern analysis and machine intelligence .
- [27] Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W., 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). URL: <https://github.com/kenziyuliu/ms-g3d>.
- [28] Maselli, M.V., Marani, R., Cicirelli, G., D'Orazio, T., 2024. Continuous action recognition in manufacturing contexts by deep graph convolutional networks, in: Intelligent Systems Conference (IntelliSys 2023), Lecture Notes in Networks and Systems, Springer. doi:[10.1007/978-3-031-47718-8_11](https://doi.org/10.1007/978-3-031-47718-8_11).
- [29] Microsoft, 2021. Azure Kinect Body Tracking SDK. <https://learn.microsoft.com/en-us/previous-versions/azure/kinect-dk/>. Accessed May 2025.
- [30] Papers with Code, 2024a. Action recognition on ntu rgb+d 120. <https://paperswithcode.com/sota/action-recognition-in-videos-on-ntu-rgb-d-120>. Accessed: 2025-07-14.
- [31] Papers with Code, 2024b. Skeleton-based action recognition on ntu rgb+d 120. <https://paperswithcode.com/sota/skeleton-based-action-recognition-on-ntu-rgb-d-1>. Accessed: 2025-07-14.
- [32] Ragusa, F., Furnari, A., Farinella, G.M., 2021. Meccano: A multimodal egocentric dataset for human behavior understanding in the industrial-like domain, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). URL: <https://iplab.dmi.unict.it/MECCANO/>.
- [33] Reilly, D., Das, S., 2024. Just add π ! pose induced video transformers for understanding activities of daily living .
- [34] Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhanian, D., Wang, R., Yao, A., 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). URL: <https://assembly-101.github.io/>.
- [35] Shahroudy, A., Liu, J., Ng, T.T., Wang, G., 2016. Ntu rgb+d: A large scale dataset for 3d human activity analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:[10.1109/CVPR.2016.550](https://doi.org/10.1109/CVPR.2016.550).
- [36] Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12018–12027.
- [37] Shi, L., Zhang, Y., Cheng, J., Lu, H., 2020. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Transactions on Image Processing 29, 9532–9545. doi:[10.1109/TIP.2020.3001125](https://doi.org/10.1109/TIP.2020.3001125).
- [38] Wang, L., Huynh, D.Q., Koniusz, P., 2019. A comparative review of recent kinect-based action recognition algorithms. [arXiv:1906.09955](https://arxiv.org/abs/1906.09955).
- [39] Xiang, L., Wang, Z., 2025. Joint mixing data augmentation for skeleton-based action recognition. ACM Transactions on Multimedia Computing, Communications, and Applications 21. URL: <https://doi.org/10.1145/3700878>, doi:[10.1145/3700878](https://doi.org/10.1145/3700878).
- [40] Xin, W., Miao, Q., Liu, Y., Liu, R., Pun, C.M., Shi, C., 2023. Skeleton mixformer: Multivariate topology representation for skeleton-based action recognition, in: Proceedings of the 31st ACM International Conference on Multimedia (MM '23), ACM, Ottawa, ON, Canada. doi:[10.1145/3581783.3611900](https://doi.org/10.1145/3581783.3611900).
- [41] Xiong, Q., Zhang, J., Wang, P., Liu, D., Gao, R.X., 2020. Transferable two-stream convolutional neural network for human action recognition. Journal of Manufacturing Systems doi:[10.1016/j.jmsy.2020.04.007](https://doi.org/10.1016/j.jmsy.2020.04.007).
- [42] Yan, S., Xiong, Y., Lin, D., 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition, in: AAAI Conference on Artificial Intelligence. URL: <https://github.com/yysijie/st-gcn>.
- [43] Zheng, H., Lee, R., Lu, Y., 2023. Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding. [arXiv preprint arXiv:2307.05721](https://arxiv.org/abs/2307.05721) .
- [44] Zhou, Y., Cheng, Z.Q., Li, C., Fang, Y., Geng, Y., Xie, X., Keuper, M., 2023. Hypergraph transformer for skeleton-based action recognition. [arXiv:2211.09590](https://arxiv.org/abs/2211.09590).
- [45] Özyer, T., Ak, D.S., Alhaji, R., 2021. Human action recognition approaches with video datasets – a survey. Knowledge-Based Systems doi:[10.1016/j.knosys.2021.106994](https://doi.org/10.1016/j.knosys.2021.106994).