

# Graph Convolutional Networks Skeleton-based Action Recognition for Continuous Data Stream: A Sliding Window Approach

Mickael Delamare<sup>1,2</sup><sup>a</sup>, Cyril Laville<sup>1</sup><sup>b</sup>, Adnane Cabani<sup>2</sup><sup>c</sup> and Houcine Chafouk<sup>2</sup><sup>d</sup>

<sup>1</sup>SIATEch SAS, 73 Rue Martainville, 76000 Rouen, France

<sup>2</sup>Normandie Univ., UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France

**Keywords:** Spatial-temporal Graph Convolutional Networks, Sliding Window, Action Recognition, Skeleton Data.

**Abstract:** This paper introduces a novel deep learning-based approach to human action recognition. The method consists of a Spatio-Temporal Graph Convolutional Network operating in real-time thanks to a sliding window approach. The proposed architecture consists of a fixed window for training, validation, and test process with a Spatio-Temporal-Graph Convolutional Network for skeleton-based action recognition. We evaluate our architecture on two available datasets of common continuous stream action recognition, the Online Action Detection dataset, and UOW Online Action 3D datasets. This method is utilized for temporal detection and classification of the performed action recognition in real-time.


## 1 INTRODUCTION


Real-time human action recognition from skeleton data streams is a central point in several applications as it allows for seamless coordination between humans and machine and can be used to improve the safety of the workplace by checking for falls or dangerous situations (Ni et al., 2020). However, this is a challenging task as the algorithm need to be able to detect the start and end of each action without any pauses between actions as well as differentiating between each action all in real-time (Li et al., 2016). The most conventional approach to this problem consists of two algorithms working together, one algorithm detects when an action is being performed then the other algorithm differentiates between all the different actions (Lara and Labrador, 2012), however, this problem requires both algorithms to work in parallel increasing computational cost and lowering accuracy overall. This method also relies on pauses between movements and was never tested on representative actions.


Online action recognition is quickly developed in recent years. It aims to locate the action segment with the partially observed action sequences, which


can be applied in real-time. Action detection algorithms are divided into two sub-parts: off-line action detection and on-line action detection. For off-line action detection, in our article for us, segmented detection is an offline training and then an online detection with this pre-training. Online action detection, in our article, means the detection in real-time and for our method, it is also the training phase using none segmented data but continuous data stream. Most of the works (Lei and Todorovic, 2018) (Nguyen et al., 2018) only consider RGB images as the input because RGB data directly reflect original information, like human posture, object pose, etc. However, the RGB input data always requires a huge amount of calculation, which is usually accelerated by GPU. Another type of input is based on skeleton data, that requires less amount of calculation and can be extracted from video or directly provided by Inertial Measurement Unit (IMU) (Polfreman, 2018) which is more convenient for workers who might move a lot and end up out of the field of vision of the camera.

However, for off-line action detection, future action is still unavailable for practical use. We propose another solution to this problem based on Spatial-Temporal Graph Convolutional Network (ST-GCN) (Yan et al., 2018), ST-GCN will be used to construct a set of spatial-temporal graph convolutions on the skeleton sequences and can capture motion information in dynamic skeleton sequence that feet our aim to detect action recognition in real-time. This ST-GCN

<sup>a</sup> <https://orcid.org/0000-0003-0119-2326>

<sup>b</sup> <https://orcid.org/0000-0002-7871-843X>

<sup>c</sup> <https://orcid.org/0000-0001-5948-8950>

<sup>d</sup> <https://orcid.org/0000-0002-6683-0010>

will operate in real-time thanks to a sliding window approach which will allow us to recognize the actions without having two algorithms to detect the beginning and the end of action but to recognize each action in real-time on a whole sequence.

We will test this approach on two datasets: UOW OnlineAction 3D dataset (Tang et al., 2018) and the OAD dataset (Li et al., 2016). Our contributions in this paper are:

- A novel sliding windows ST-GCN based approach for human action recognition.
- A showcases how effective this approach is compared to the state of the art on two datasets: UOW OnlineAction 3D dataset and Online Action Detection which uses representative challenging actions.

The rest of the paper is outlined as follows. Section 2 describes related works of motion action recognition. The SW-GCN method is presented in Section 3, and experimental results are shown in Section 4. The last section concludes this paper.

## 2 RELATED WORK

The most common approaches to human action recognition focus on classifying different actions on segmented data streams (Mitra and Acharya, 2007), where the classifier is provided with individual manually segmented actions and only has to identify which action is being performed. Hidden Markov's models are often used for this purpose (Tao et al., 2012) but they are slow and require a large dataset. However, one major limitation of this approach is the fact that to apply it in a real-time scenario another algorithm is required to segment the data stream (Zhao et al., 2013). The addition of another algorithm increases the complexity of the system, adds another source of potential errors, and is computationally expensive.

### 2.1 Segmented Action Recognition

Most of the existing approaches for skeletons based action recognition model the spatial-temporal evolution of actions based on hand-crafted features. As a kind of hierarchically adaptive filter bank, CNN performs well in representation learning. An end-to-end hierarchical architecture for skeletons based action recognition with CNN has been proposed (Du et al., 2015).

A fast and highly accurate action recognition system based on Long Short Term Memory (LSTM) and CNN that are trained to process input sequences of 3D

hand positions and velocities acquired from infrared sensors for recognition of dynamic Hand actions has been proposed (Naguri and Bunescu, 2017).

They showed Segmented motion action was really accurate for each segmentation, but in most of the case they need at least two algorithms for continuous stream action recognition and the accuracy is degraded due to online detection or segmentation. This is why we focus on online motion action with one algorithm.

### 2.2 Continuous Online Action Recognition

A continuous Hidden Markov Model for online action recognition based on vision has been introduced (Eickeler et al., 1998). The system is able to recognize dynamic actions in person and background-independent mode and works several times faster than real-time. A method based on Hidden Markov Models (HMMs) presented for dynamic action trajectory modeling and recognition has been proposed (Wang et al., 2012). An online version of the expectation-maximization (EM) algorithm for HMMs has been presented (Mongillo and Deneve, 2008). The online algorithm is able to deal with dynamic environments when the statistics of the observed data is changing with time. The HMM method is the first method used for online action recognition.

An approach that dynamically adjusts the window size and the shift at every step has been proposed (Laguna et al., 2011). One limitation is instances depend on the accuracy of the sensors. If sensors do not capture a significant change in the environment, the system does not detect the state change and it does not create the corresponding instance. It's why we chose to use a fixed window with skeleton data in our context.

A method for real-time action recognition from a noisy skeleton stream, such as those extracted from Kinect depth sensors has been introduced (Miranda et al., 2014). This method can improve the input of GCN.

An online dynamic hand action recognition system with an RGB-D camera, which can automatically recognize hand actions against the complicated background is presented (Xu et al., 2015).

The authors (Molchanov et al., 2016) employ connectionist temporal classification to train the network to predict class labels from in-progress actions in unsegmented input streams. This method provides an Online detection and classification of dynamic hand actions with recurrent 3d convolutional neural networks.

A sliding window approach to data processing is used (Luzhnica et al., 2016), their algorithm is suitable for stream data processing for natural hand action recognition.

### 2.3 Survey on Spatial-temporal Graph Convolutional Networks

The authors (Wu et al., 2020) provide a taxonomy that groups neural networks of graphs into four categories: neural networks of recurrent graphs, neural networks of convolutional graphs, graphauto-coders, and neural networks of space-time graphs.

The noisy skeleton-based action recognition method based on convolutional graph networks with predictive coding for latent space called predicatively coded convolutional graph networks (PeGCN) is presented (Yu et al., 2020). It increases the flexibility of the GCN and is better suited for action recognition tasks using skeletal characteristics. This paper also strengthens our choice by using skeleton data. A novel Attention Enhanced Graph Convolutional LSTM Network (AGC-LSTM) for human action recognition from skeleton data is proposed (Si et al., 2019). The proposed AGC-LSTM can not only capture discriminative features in spatial configuration and temporal dynamics but also explore the co-occurrence relationship between spatial and temporal domains. A novel two-stream adaptive graph convolutional network (2s-AGCN) for skeleton-based action recognition is presented (Shi et al., 2019). This data-driven approach increases the flexibility of the graph convolutional network and is more suitable for the action recognition task.

A novel model of dynamic skeletons called ST-GCN is proposed (Yan et al., 2018), which moves beyond the limitations of previous methods by automatically learning both the spatial and temporal patterns from data. This formulation not only leads to greater expressive power but also stronger generalization capability.

The ST-GCN for skeleton-based action recognition is extended by introducing two novel modules, namely, the GraphVertex Feature Encoder (GVFE) learns appropriate vertex features for action recognition by encoding raw skeleton data into a new feature space. And the Dilated Hierarchical Temporal Convolutional Network (DH-TCN) is capable of capturing both short-term and long-term temporal dependencies using a hierarchical dilated convolutional network (Papadopoulos et al., 2019). To capture richer dependencies, (Li et al., 2019) introduce an encoder-decoder structure, called A-link inference module, to capture action-specific latent dependencies directly

from actions. They also extend the existing skeleton graphs to represent higher-order dependencies. The authors (Zheng et al., 2019) shows that the model has high robustness and accuracy. BVH data is used which are skeleton data, using the ST-GCN algorithm. This paper shows that skeleton data and ST-GCN is efficient and strengthen our choice using ST-GCN.

The Graph convolution network is a recent approach and shows its effectiveness as mentioned above, to detect actions with skeleton data. We choose this ST-GCN algorithm (Yan et al., 2018) to provide action recognition with a sliding windows approach to be able to detect motion action in real-time and only focus on the sliding windows instead of improving the ST-GCN. Other authors have improved this ST-GCN such as (Li et al., 2019),(Papadopoulos et al., 2019),(Zheng et al., 2019) or (Zheng et al., 2019)

### 2.4 Survey on Sliding Window Approach

The authors (Laguna et al., 2011) used a different approach using dynamic windows based on events. Their approach dynamically adjusts the window size and the shift at every step. Experiments with public datasets show that their method, employing simpler models, is able to accurately recognize the activities.

Overlapping sliding windows in Human action Recognition (HAR) systems are associated with underlying limitations of subject-dependent cross-validation (CV). When a subject independent CV is used, overlapping sliding windows do not improve the performance of HAR systems but nevertheless require substantially more resources than non-overlapping windows (Dehghani et al., 2019). We choose overlapping sliding windows in our context to have more data to characterize, and the algorithm can be updated more frequently.

Determining the start time and end time of the action increases the computation load, so the recognition results will be delayed (Ma et al., 2020). That's why we chose the sliding window method to recognize the actions with their surrounding noise without beginning and end recognized in our algorithm. This would decrease the computation load and could be deployed in an embedded system.

## 3 THE SW-GCN METHOD

### 3.1 A Sliding Window Approach

In the last decade, the theoretical study of the sliding window model was developed to advance applications

with very large input and time-sensitive output. In some practical situations, the input might be seen as an ordered sequence, and it is useful to restrict computations to recent portions of the input, (Datar et al., 2002) introduced the sliding window model that assumes that the in real-time. To validate our method, on Figure 9 test was made, and on Figure 10, the testing input is a stream of data elements and divides the data elements into two categories: active elements and expired elements. We denote the stream  $D$  by a sequence of elements  $\{P_i\}_{i=1}^m$  where  $p_i \in \mathbb{N}$ . It is important to note that  $m$  is incremented for each new arrival. For a subset  $Z_j$  of the state space,  $j \in 1, \dots, r$ , for all  $x \in Z_j$ . The sets  $Z_1, \dots, Z_r$  are called windows, and we assume that  $Z_j$  contains actions during the time interval  $[t_{j-1}, t_j]$ . It is assumed that the size of the window plays a crucial role in this method. This is why we choose a fixed window size that corresponds to the size of an average action on each dataset. For labeling, we take the middle frame to define the action corresponding to the sliding window as shown in Figure 1. This will allow the algorithm to characterize the pieces of actions for each window with the measurement noise induced by the other actions. The space-time graph is built on the skeleton sequences in two steps. First, the joints within a frame are connected by edges according to the connectivity of the human body structure. Then, each articulation will be connected to the same articulation in the consecutive frame. The connections in this configuration are therefore naturally defined without the manual assignment of parts. This also allows the network architecture to work on data sets with a different amount of articulation or joint connectivity (Yan et al., 2018). Within the sliding window method combined with ST-GCN, it can capture motion information in dynamic skeleton sequences in real-time.

Our sliding window is fixed at the same size during the training phase and the testing phase. The size is determined by the average length of action for each dataset shown in Figure 1. The offset of the sliding window is one frame by one frame.

### 3.2 Spatial Temporal Graph Convolutional Network

ST-GCN is a neural network that takes skeletal data as an input and uses a Spatio-temporal kernel to detect movements in the skeleton. This allows the network to detect and classify different actions without the need for a heavy algorithm.

This is why we decided to choose the Spatial-Temporal Graph Convolutional Network to detect action and characterize noise around the action using

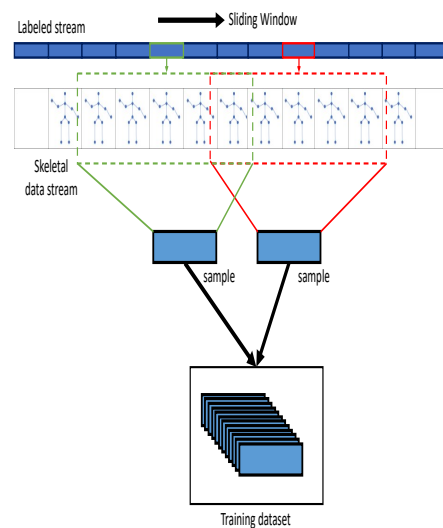


Figure 1: Pre-processing of the Skeleton layout with each joint used and Labeled Sliding window.

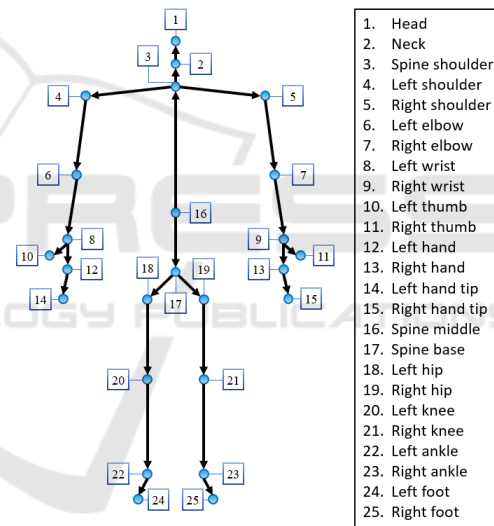


Figure 2: Skeleton layout with each joint used for both dataset (25 joints).

the sliding windows. Only skeletal data were used because skeletal data can be obtained by inertial sensors which are the cheapest on the market compared to the camera or motion capture system. In our point of view, in an industrial context, this is the best choice.

## 4 EXPERIMENTS

The proposed method is evaluated on two challenging datasets: the Online Action Detection dataset (Li et al., 2016) and the UOW Online Action 3D dataset (Tang et al., 2018) that have whole and unsegmented

Table 1: Result of UOW Online Action 3D action Comparison between CNN and GCN.

Method	Accuracy	F1-score
SW-CNN	0.680	0.680
<b>SW-GCN</b>	<b>0.755</b>	<b>0.750</b>

online sequences. In all the datasets, multiple actions are contained in sequences of videos with skeleton data of which we only used the latter. The skeleton layout with each joint used is organized as shown in Figure 2. Both datasets using 25 joints so we used the same input graph adjacency matrix for the ST-GCN.

We used the F1-score and accuracy to determine a correct detection. A detection is correct when the labeled frame in the middle of the sliding window matches the prediction. We have not used SL-score and EI-score, our aim is to detect action without knowing the start and end of an action.

#### 4.1 Experiments on the UOW Online Action 3D Dataset

The UOW Online Action 3D dataset (Tang et al., 2018) contains 20 different actions, performed by 20 different subjects with up to 3-5 different executions. From each of the 48 sequences, the 25 joint positions per frame were used as inputs.

We choose this dataset instead of the MSR 3D action dataset (Li et al., 2010) because it contains the same actions but it has continuous sequences of actions which are our aim.

The UOW Online Action 3D dataset is recent and does not propose a method similar to ours so we have created a Sliding Windows based on Convolutional Neural Network (SW-CNN). To compare with the SW-GCN method. The SW-CNN was trained under the ranger optimizer, which consists of two components: Rectified Adam (RAdam) and Lookahead (Liu et al., 2019b), for 200 epochs. And consists of four convolution layers with 40 to 160 filters, as well as 2 max-pooling layers followed by a fully connected layer of 100 neurons and the Mish activation function (Misra, 2020). The loss function used was a weighted cross-entropy loss function. This demonstrates the efficiency of the GCN algorithm through the sliding window which is the same as both methods. The SW-GCN was trained under Stochastic gradient descent for 140 epochs and consists of 11 layers with 32 to 128 filters and a reluctant linear activation function. The loss function used was a weighted cross-entropy loss function.

The data have been reorganized into windows of 50 frames as it is the average duration of all actions in this dataset, with the method previously shown, then

each joint has been separated and the values have been re-centered around zero, before being divided by three to put them between -1 and 1. Finally, the data have been separated into a training set of 46 sequences, a validation set of 1 sequence, and a testing set of 1 sequence.

This dataset allows us to show the effectiveness of our new approach. The SW-GCN has an F1-score of 0.75 while the SW-CNN has 0.68 seen in the table 1. However, the SW-CNN is 10 times faster than our method (1.63ms instead of 10ms) but using the CNN can have a more false-positive prediction because inputs are just a simple matrix. Using the ST-GCN provides information on inputs as the skeleton matrix.

#### 4.2 Experiments on the OAD Dataset

The OAD dataset (Li et al., 2016) contain long sequences corresponding to 700 action sequences with ten action classes collected with Kinect v2. The data have been reorganized into windows of 50 frames as it is the average duration of all actions in this dataset, with the method previously shown, then each joint has been separated and the values have been re-centered around zero, before being divided by three to put them between -1 and 1. Finally, the data have been separated into a training set of 46 sequences, a validation set of 1 sequence, and a testing set of 1 sequence.

The ST-GCN was trained under Stochastic gradient descent for 140 epochs and consists of 11 layers with 32 to 128 filters and a reluctant linear activation function. The loss function used was a weighted cross-entropy loss function.

The authors (Liu et al., 2019a) obtained 0.82 overall accuracies, with our method we have better accuracy at 0.90 for online action detection overall. We have better accuracy comparing the F1-score of each action except for only one action (Drinking). This can be explained by the fact that the action Drinking is not well recognized by our method and cannot be detected around the noise this is mostly due to the window size. The action is too short compared to the size of the window and is not recognized compared to other actions that are longer.

#### 4.3 Evaluation of Our Method

The measurements were realized on a laptop with an i7-8750H processor and a GTX 1070. For both datasets, we measured both inference time and throughput of the network. Inference time was measured after a GPU warms up and was measured for 300 repetitions. The average inference time on the UOW OnlineAction 3D Dataset was 10.1 ms and the

Table 2: Comparison on the OAD Dataset F1-Score for each class.

Actions	SVM-SW (Li et al., 2016)	RNN-SW (Zhu et al., 2016)	CA RNN (Li et al., 2016)	JCR RNN (Li et al., 2016)	SW-GCN (Our method)
Drinking	0.15	0.44	<b>0.58</b>	0.57	0.09
Eating	0.47	0.55	0.56	0.52	<b>0.84</b>
Writing	0.65	0.86	0.75	0.82	<b>0.92</b>
Opening cupboard	0.30	0.32	0.49	0.50	<b>0.89</b>
Washing hands	0.56	0.67	0.67	0.71	<b>0.78</b>
Opening Microwave	0.60	0.67	0.47	0.70	<b>0.78</b>
Sweeping	0.46	0.59	0.60	0.64	<b>0.93</b>
Gargling	0.44	0.55	0.58	0.62	<b>0.95</b>
Trowing trash	0.55	0.674	0.43	0.46	<b>0.88</b>
Wiping	0.86	0.75	0.76	0.78	<b>0.96</b>

Table 3: Comparison on the OAD Dataset Accuracy overall.

Method	Accuracy
ST-LSTM (Liu et al., 2017a)	0.77
AttentionNet (Liu et al., 2017b)	0.75
JCR-RNN (Li et al., 2016)	0.79
FSNet (Liu et al., 2019a)	0.80
SSNet (Liu et al., 2019a)	0.82
<b>SW-GCN (Our method)</b>	<b>0.90</b>

average inference time on the Online Action Detection dataset was 11.2 ms. We do not use any data augmentation when training models.

The throughput of the network was measured over one second and was of 2544 repetitions for the UOW OnlineAction 3D dataset and of 2515 repetitions for the Online Action Detection dataset. The inference time for both datasets is about 10 ms that is acceptable for online action recognition in real-time.

We obtained good results with the OAD dataset in online action recognition shown in Table 3 and Table 2. Our method has better result accuracy than the SW-CNN of the UOW Online Action 3D dataset shown in Table 1 which proves that our method can generalize whole sequences of action recognition. The algorithm is capable of characterized an action even if the action is noisy.

To show the aim of our action recognition method, the result of the validation is seen in Figure 3 for OAD dataset and in Figure 4 that show the whole predictions of the validation, sequence compare with

the ground truth sequence with 90% accuracy. It is a graph of all detected actions in real-time. To validate our method, on Figure 5 test sequence was made and we had 91% accuracy. On Figure 6, the testing phase that fit the ground truth sequence. The class list is [0: No action, 1: Drinking, 2: Eating, 3: Writing, 4: Opening Cupboard, 5: Washing hand, 6: Opening Microwave Oven, 7: Sweeping, 8: Gargling, 9: Throwing trash, 10: Wiping].

For the UOW dataset, results of the validation are seen in Figure 7 for OAD dataset and on Figure 8 that show the whole predictions of the validation sequence compare with the ground truth sequence with 75% accuracy. It is a graph of all detected actions in real-time. To validate our method, on Figure 9 test was made, and on Figure 10, the testing phase that fit the ground truth sequence in green with 73% accuracy. The class list is [0: No action, 1: High arm wave, 2: Horizontal arm wave, 3: Hammer, 4: Hand catch, 5: Forward punch, 6: High throw, 7: Draw X, 8: Draw tick, 9: Draw circle, 10: Hand clap, 11: Two hands wave, 12: Side boxing, 13: Bend, 14: Forward kick, 15: Sidekick, 16: Jogging, 17: Tennis swing, 18: Tennis serve, 19: Golf swing, 20: Pick up and throw].

For both datasets, we have a good representation of continuous action recognition in real-time. We also produced a video sequence to show in real-time our SW-GCN solution. All the confusion matrix are available in the Github repository as well as the code to reproduce our method at : <https://github.com/DelamareMicka/SW-GCN>.

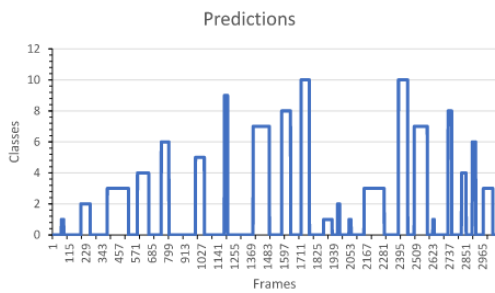


Figure 3: Predictions of SW-GCN method with validation sequence in blue for OAD dataset with 90% accuracy.

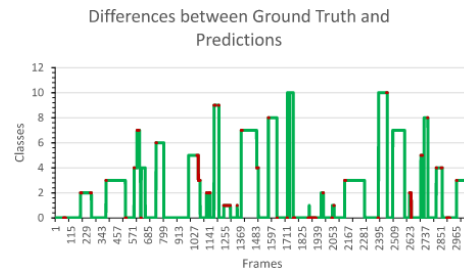


Figure 4: Predictions errors of SW-GCN method highlighted in red with validation sequence for OAD dataset. In green the predictions that corresponds to the ground truth.

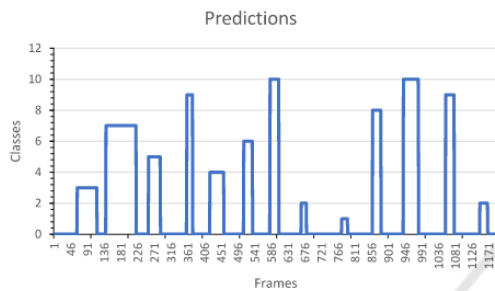


Figure 5: Predictions of SW-GCN method with test sequence in blue for OAD dataset with 91% accuracy.

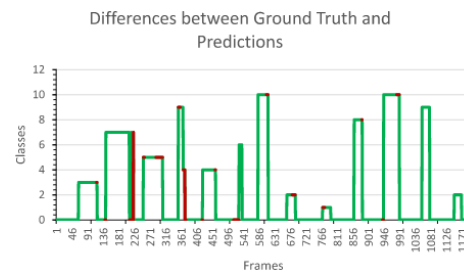


Figure 6: Predictions errors of SW-GCN method highlighted in red with validation sequence for OAD dataset. In green the predictions that corresponds to the ground truth.

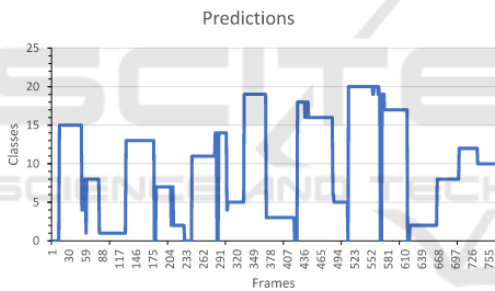


Figure 7: Predictions of SW-GCN method with validation sequence in blue for UOW dataset with 75% accuracy.

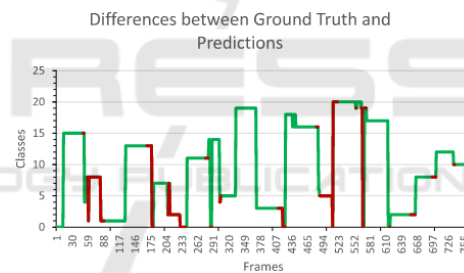


Figure 8: Predictions errors of SW-GCN method highlighted in red with validation sequence for UOW dataset. In green the predictions that corresponds to the ground truth.

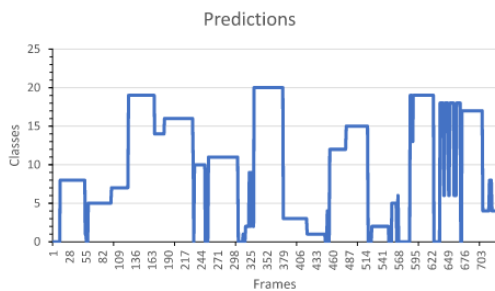


Figure 9: Predictions of SW-GCN method with test sequence in blue for UOW dataset with 73% accuracy.

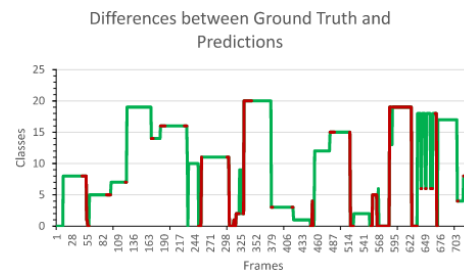


Figure 10: Predictions errors of SW-GCN method highlighted in red with test sequence for UOW dataset. In green the predictions that corresponds to the ground truth.

## 5 CONCLUSION AND FUTURE DIRECTIONS

In this paper, it has been shown that the sliding window approach coupled with the Spatial-Temporal Graph Convolutional Networks allows taking advantage of it since the Graph convolutional Network uses temporal information of the skeleton and can characterize the noise around the action to determine the right action in the sliding windows. We have shown a sliding window is a good approach for online action recognition in real-time with continuous data streams, and it does not require a powerful processor like two algorithms, one for data stream segmentation, a second for action recognition. Our method provides only one algorithm. And it can be embedded in a small Electronic Control Unit (ECU) to provide a fast inference of the current action.

One of the limits is the size of the sliding and effective window when we know the average duration of action. We validate our method with two states of the art data sets with a common real-time motion action and have shown a good performance.

Our future works will focus on a variable sliding window that allows knowing several actions with different lengths. The main challenging is the amount of data, the InHard dataset (Dallel et al., 2020) correspond with the aim of action recognition in industrial sites. But it needs much more data to generalized action detection, it will be a part of our work to enlarge this dataset. Our method can also be improved by using improved ST-GCN shown in the survey on ST-GCN.

## REFERENCES

- Dallel, M., Havard, V., Baudry, D., and Savatier, X. (2020). Inhard - an industrial human action recognition dataset in the context of industrial collaborative robotics. In *IEEE International Conference on Human-Machine Systems ICHMS*.
- Datar, M., Gionis, A., Indyk, P., and Motwani, R. (2002). Maintaining stream statistics over sliding windows. *SIAM journal on computing*, 31(6):1794–1813.
- Dehghani, A., Sarbishei, O., Glatard, T., and Shihab, E. (2019). A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors. *Sensors*, 19(22):5026.
- Du, Y., Fu, Y., and Wang, L. (2015). Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583. IEEE.
- Eickeler, S., Kosmala, A., and Rigoll, G. (1998). Hidden markov model based continuous online gesture recognition. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 2, pages 1206–1208. IEEE.
- Laguna, J. O., Olaya, A. G., and Borrajo, D. (2011). A dynamic sliding window approach for activity recognition. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 219–230. Springer.
- Lara, O. D. and Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209.
- Lei, P. and Todorovic, S. (2018). Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6742–6751.
- Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3595–3603.
- Li, W., Zhang, Z., and Liu, Z. (2010). Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 9–14. IEEE.
- Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., and Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. In *European Conference on Computer Vision*, pages 203–220. Springer.
- Liu, J., Shahroudy, A., Wang, G., Duan, L.-Y., and Kot, A. C. (2019a). Skeleton-based online action prediction using scale selection network. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1453–1467.
- Liu, J., Shahroudy, A., Xu, D., Kot, A. C., and Wang, G. (2017a). Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021.
- Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. (2017b). Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1656.
- Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019b). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.
- Luzhnica, G., Simon, J., Lex, E., and Pammer, V. (2016). A sliding window approach to natural hand gesture recognition using a custom data glove. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 81–90. IEEE.
- Ma, C., Li, W., Cao, J., Du, J., Li, Q., and Gravina, R. (2020). Adaptive sliding window based activity recognition for assisted livings. *Information Fusion*, 53:55–65.
- Miranda, L., Vieira, T., Martínez, D., Lewiner, T., Vieira, A. W., and Campos, M. F. (2014). Online gesture



- recognition from pose kernel learning and decision forests. *Pattern Recognition Letters*, 39:65–73.
- Misra, D. (2020). Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, pages 1–14.
- Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324.
- Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., and Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4207–4215.
- Mongillo, G. and Deneve, S. (2008). Online learning with hidden markov models. *Neural computation*, 20(7):1706–1716.
- Naguri, C. R. and Bunesco, R. C. (2017). Recognition of dynamic hand gestures from 3d motion data using lstm and cnn architectures. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1130–1133. IEEE.
- Nguyen, P., Liu, T., Prasad, G., and Han, B. (2018). Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761.
- Ni, P., Lv, S., Zhu, X., Cao, Q., and Zhang, W. (2020). A light-weight on-line action detection with hand trajectories for industrial surveillance. *Digital Communications and Networks*.
- Papadopoulos, K., Ghorbel, E., Aouada, D., and Ottersten, B. (2019). Vertex feature encoding and hierarchical temporal modeling in a spatial-temporal graph convolutional network for action recognition. *arXiv preprint arXiv:1912.09745*.
- Polfreman, R. (2018). Hand posture recognition: Ir, semg and imu.
- Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035.
- Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. (2019). An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1227–1236.
- Tang, C., Li, W., Wang, P., and Wang, L. (2018). Online human action recognition based on incremental learning of weighted covariance descriptors. *Information Sciences*, 467:219–237.
- Tao, L., Elhamifar, E., Khudanpur, S., Hager, G. D., and Vidal, R. (2012). Sparse hidden markov models for surgical gesture classification and skill evaluation. In *International conference on information processing in computer-assisted interventions*, pages 167–177. Springer.
- Wang, X., Xia, M., Cai, H., Gao, Y., and Cattani, C. (2012). Hidden-markov-models-based dynamic hand gesture recognition. *Mathematical Problems in Engineering*, 2012.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.
- Xu, D., Wu, X., Chen, Y.-L., and Xu, Y. (2015). Online dynamic gesture recognition for human robot interaction. *Journal of Intelligent & Robotic Systems*, 77(3-4):583–596.
- Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.
- Yu, J., Yoon, Y., and Jeon, M. (2020). Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition. *arXiv preprint arXiv:2003.07514*.
- Zhao, X., Li, X., Pang, C., Zhu, X., and Sheng, Q. Z. (2013). Online human gesture recognition from motion data streams. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 23–32.
- Zheng, W., Jing, P., and Xu, Q. (2019). Action recognition based on spatial temporal graph convolutional networks. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, pages 1–5.
- Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. *arXiv preprint arXiv:1603.07772*.