Body Gestures Recognition for Social Human-Robot Interaction

Javier Laplaza, Joan Jaume Oliver Caraballo, Alberto Sanfeliu and Anaís Garrell
Universitat Politècnica de Catalunya - BarcelonaTech (UPC)
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain
{javier.laplaza, joan.jaume.olive, alberto.sanfeliu, anais.garrell}@upc.edu

***Abstract*—In this paper, a solution for human gesture classification is proposed. The solution uses a Deep Learning model and is meant to be useful for non-verbal communication between humans and robots. The research focuses on the creation of the "temPoral bOdy geSTUre REcognition model" (POSTURE) that can recognise continuous gestures performed in real-life situations. The suggested model takes into account spatial and temporal components so as to achieve the recognition of more natural and intuitive gestures. In a first step, a framework extracts from all the images the corresponding landmarks for each of the body joints. Next, some data filtering techniques are applied with the aim of avoiding problems related with the data. And finally, different neural network configurations and approaches are tested to find the optimal performance. The validation of the model is accomplished throughout an extensive set of simulations and real-life experiments.**

***Index Terms*—Human Robot Interaction, gesture recognition.**

I. Introduction

While languages have evolved as the primary tools for human communication, nonverbal communication remains profoundly influential. Extensive research over the years has established that gestures and body language are the principal nonverbal modes through which humans convey substantial amounts of information.

Gestures, in essence, are deliberate actions performed with parts of the body to communicate distinct messages. When employed effectively, gestures and body language can create impressions and or capture attention, underscoring their significant role in human interaction.

Initially, human-machine communication was predominantly physical, primarily involving keyboards or touchscreens [1]. However, human interaction inherently relies on the recognition and interpretation of speech, gestures, and emotions, rather than on touchscreens alone [2]. Human Robot Interaction (HRI) research focuses in a natural collaboration between humans and robots which is the key to make robotics effective enough to solve countless real-world challenges [3], [4].

Speech recognition is, therefore, one of the most convenient methods of human-machine interaction, despite challenges posed by the diversity of human accents and its susceptibility to noisy environments. Alternatively, vision-based methods are a prominent area of research, given their capacity to convey complex information effectively.

This work was supported by JST Moonshot R & D (JPMJMS2011-85), LENA Spanish national project (PID2022-142039NA-I00), and European project CANOPIES (H2020- ICT-2020-2-101016906).

Fig. 1: Predicted human gestures to communicate with a robot .

The primary motivation for developing these touchless methods is to foster engagement between humans and robots, thereby facilitating more natural interactions. Enhanced collaboration leads to improved performance and effectiveness of robotic systems. [5].

In this paper, we focus on the recognition of body gestures as a natural and intuitive means of communicating with robots. To achieve this, we propose a deep learning approach capable of recognizing body gestures in video sequences. The primary contribution of this work is the development of a deep learning model that takes video input and accurately classifies body gestures.

In the remainder of the paper we start by introducing the POSTURE model in Section II. Results are detailed in Section III. Finally, conclusions are presented in Section IV.

II. POSTURE Model

This section discloses with a bit of detail the proposed tem**P**oral b**O**dy ge**STU**re **RE**cognition (POSTURE) model. Generally speaking, the model can be divided into two main areas. The first one is the responsible of the image treatment and body landmarks extraction, while the second one is the area in charge of the output prediction.

Given a video, the model is able to predict the gesture is being shown. Each video input is 30 frames long, from which we first extract and encode the position landmarks that are next fed to the Neural Network.

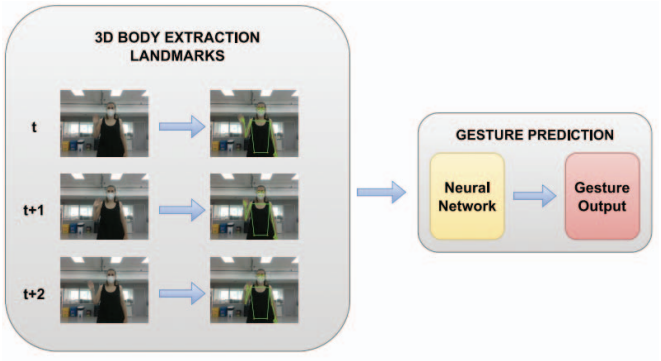

Fig. 2: System Overview.

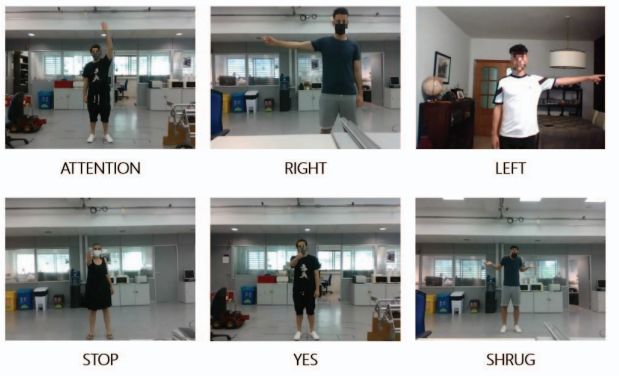

Fig. 3: Some samples of gestures recorded in the dataset.

In this case, the model used is a classifier with a densely connected network that outputs a vector containing each gesture probability, that is used to make the final decision.

A. Landmarks extraction

In order to extract the desired landmarks or body joints, MediaPipe has been the tool used. MediaPipe Pose is a machine learning solution for high-fidelity body pose tracking [6]. This framework is really helpful since it's able to properly localize landmarks under different conditions of lighting, distance and half body configurations.

In order to get these joints, Pose uses a two-step strategy. First of all, the algorithm locates the Region-Of-Interest within the image (*the body*). And afterwards, predicts the 33 landmarks using the Region-Of-Interest-cropped frame as input. In case of using videos as inputs, the system only tracks the body once and then the predicted pipeline get's the Region-Of-Interest from previous body positions.

This two-step algorithm, helps our prediction eluding the rotation, translation and scale problems while still granting a proper and stable output. Pose outputs 33 3D landmarks for each video frame that we then use to build a 33 node unidirectional graph Neural Networks

Each node is configured as a one dimensional vector with four elements, joint position x, joint position y, joint position z and joint *visibility*.

B. Body Gestures Dataset

In order to train, test and validate the proposed model, IRIGesture dataset [7] has been used. This dataset is handmade and it was created in the Mobile Robotics laboratory [8].

The main feature of this gesture based communication dictionary is naturalness. The main objective is that everyone can communicate with robots and not only people who is already familiar with them.

The dataset contains 450 sample videos (divided in 10 different subjects) of static and dynamic human gestures. Static gesture are those that require a certain amount of movement to be done and then they remain static. Dynamic gestures instead, are constantly in movement.

Static gestures

- **Attention**: Catch the robot's attention to give him an order.
- **Right**: Order the robot to turn right.
- **Left**: Order the robot to turn left.
- **Stop**: Order the robot to stop its trajectory.
- **Yes**: Approve a robot's information.
- **Shrug**: Inform the robot that you don't understand his information.
- **Random**: Random gesture, not necessarily a communicatve gesture.
- **Static**: Human is standing still.

Dynamic gestures

- **Greeting**: Greet the robot.
- **Continue**: Order the robot to continue its path after telling him to stop.
- **Turn-back**: Order the robot to turn 180 degrees.
- **No**: Deny a robot's information.
- **Slowdown**: Order the robot to reduce its speed.
- **Come**: Order the robot to reach your position.
- **Back**: Order the robot to move back.

For data recording, each human volunteer was captured using an RGB camera. Volunteers were given a vague explanation of the intended gesture to encourage natural responses. There were no restrictions on which arms to use, allowing volunteers to perform the same gesture differently, using one arm, the other, or both.

Each gesture was repeated three times: first at a distance of 1 meter from the camera, then at 4 meters, and finally at 6 meters. Each video captured only one gesture, and all recordings were conducted indoors.

1) Data Augmentation

In deep learning, a large dataset is essential for optimal model performance. To ensure our model accurately comprehends all possible behaviors, extensive training data is required. With only 450 videos (30 per class), we observed that our model struggled to fully understand the data. Therefore, we applied data augmentation tech-

niques. Data augmentation involves artificially generating new data from existing data to enhance the training dataset.

For this dataset, we applied data augmentation by using sequences of 30 frames with 3-frame gaps, generating nearly 3,000 new videos.

2) Key Nodes Selection

Out of the 33 possible joints that Mediapipe returns for each video frame, not all are useful for our purposes. Since most of our gestures involve only the hands and trunk, it was essential to analyze the impact of other joints, such as those from the face, legs, or feet, on our model. In this study, joints from the face, legs, and feet were found to be non-essential and were excluded. A filtering mask was applied, reducing the number of joints from 33 to 15.

3) Gestures Selection

As demonstrated, using all available data is not always optimal, and data filtering may be necessary. In the initial experimentation phases, we observed that while the model performed adequately on the training data, it struggled significantly with the test data. This issue stemmed from some gestures, such as those labeled as "random," being too nonspecific.

4) Key Frames Selection

Given the naturalness of the dataset, from the 450 original videos it has, none of them have the same length nor gesture speed. This implies that some gestures depending on the person who is doing them might start earlier or later, while others could be fast or slow. Of all these issues, some are already solved by the Data Augmentation techniques used (section II-B1), but the problems that refer to the start gesture trigger are still present.

5) Class Imbalance

Initially, our dataset had an equal number of videos per class, ensuring a balanced distribution. However, after applying data augmentation techniques, the total number of videos increased, leading to an imbalance in the number of videos per gesture due to variations in the original video lengths.

To address potential issues arising from this class imbalance, we adjusted the loss computation and backpropagation based on the gesture proportions. Specifically, we calculated the percentage of each gesture after data augmentation and used these proportions to weight the loss function. This approach ensures that more populous classes are penalized appropriately, promoting balance and preventing unintended biases towards certain gestures.

6) Key Edges Selection

We do not want to fix and determine the main relations between nodes, but give total freedom to try and define them as the Neural Network considers more appropriate. Since we are applying spatial attention, we opted to create a highly connected graph where every node is interconnected with all other nodes, rather than using a graph based on natural human body connections. Our goal is not to predefine specific relationships between nodes but to allow the neural network the flexibility to identify and establish these relationships as it deems most appropriate.

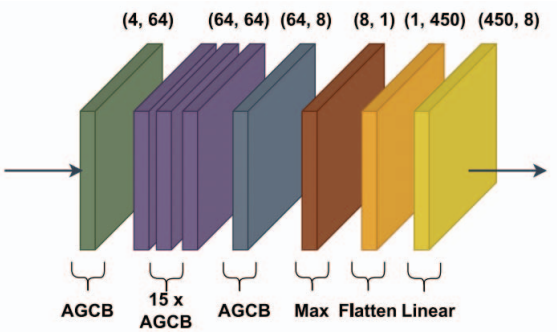

Fig. 4: Proposed Model.

7) Neural Network Architecture

Adaptive Graph Attention Networks (AAGCN) [9] are a State of the Art development that focuses on human body gestures recognition and it's based on *SpatioTemporal Graph Convolution Networks* [10]. The model uses Spatial and Temporal attention. (Encoded inside the Adaptative Graph Convolutional Block (AGCB)).

The proposed model (see Fig. 4) we have used the AGCB layer a total of 17 times combined with a few other blocks. In the first AGCB layer, our model increases the number of features per graph from 4 to 64. Once we have 64 features per graph, we keep this number static and apply the AGCB 15 times more. After we have had all these transformations, we decrease the number of features down to 8. With this, the model returns 8 features per node and frame, as if all of them were being classified individually.

Subsequently, we apply a max layer which, of all this features extracts the maximum values of each node and frame. Returning a unique graph that's a result of the mixture of the 8 graphs the last AGCB layer has returned. With only one graph, we start the prediction part of the model, in which we flatten all features of all nodes and frames.

With 450 features initially, we applied a linear layer to reduce the dimensionality to 8, corresponding to the number of gestures we aim to classify. For the propagation process, we utilized Cross-Entropy as the loss function and Adam as the optimizer.

Cross-Entropy is a widely used loss function in classification tasks due to its effectiveness in measuring the performance of classification models. Adam (Adaptive Moment Estimation) was chosen as the optimizer because it combines the advantages of two gradient descent techniques: momentum and Root Mean Square Propagation (RMSProp). In addition to the optimizer,

we implemented a learning rate scheduler that reduces the learning rate by 5% every 100 epochs to enhance the network's performance and convergence.

III. Experimentation and Results

In this section, all the refinement done to the POSTURE model for the purpose of achieving better performance will be detailed.

A. Model Experiments

In the initial stage of experimentation, several decisions, such as Key Nodes Selection, were made concurrently with model development. As a result, these decisions were well-defined by the time the experimentation phase began. Conversely, decisions regarding Key Frames Selection and the configuration of Hidden Layers were more complex and necessitated additional time and experimentation to determine.

- **Gestures Selection**

After excluding poorly defined and ambiguous gestures such as *random*, *static*, and *back*, we proceeded with experiments to analyze the behavior of the remaining gestures.

Experiment 1:

The initial test was conducted using 15 hidden layers and focused solely on the static gestures (*attention*, *right*, *left*, *stop*, *yes*, and *shrug*). This configuration achieved an accuracy of approximately 82% on the test dataset.

Experiment 2:

With such an optimal algorithm, the next step in the list was to repeat the test, but with all the gestures instead. Dynamic (*greeting*, *continue*, *turnback*, *no*, *slowdown* and *come*) plus Static ones. This second experiment, didn't go as good as we expected and the accuracy over the test dataset dropped nearly to 54%.

Experiment 3:

Since the model did not perform well with all gestures combined, the next step was to evaluate it using only dynamic gestures. With only dynamic gestures, the model achieved an accuracy of 62%. Although this was lower than the accuracy with static gestures alone, it was an improvement over the overall result. This indicates that the model performs better when dynamic and static gestures are classified independently.

Examining the confusion matrix for all gestures (see Table I), we observe that the model performs exceptionally well with gestures such as *attention*, *right*, and *left*. However, it struggles with other gestures like *continue*, *come*, and *slowdown*.

- **Hidden Layers**

Experiment 4:

Once the gestures to be used were determined, the next step was to explore whether increasing the number of layers in the neural network would enhance accuracy.

This experiment focused on static gestures, using a configuration with 25 hidden layers. Contrary to expectations, the accuracy decreased slightly rather than improving. With this setup, the model achieved an accuracy of approximately 70%.

Hidden Layers	Accuracy
15	82 %
25	70 %

TABLE II: POSTURE results with different number of hidden layers.

Up to this point, we concluded that increasing the number of hidden layers does not imply an increase in Test Accuracy.

- **Key Frames Selection**

Experiment 5:

After encountering difficulties in improving the model through both active and passive approaches, a brief investigation into dataset quality revealed several issues. It was found that the number of frames and the starting points of gestures varied across subjects and videos.

In some videos, gestures began at 0.1 seconds, while in others, they started after the first second. As a result, a time trigger of 0.7 seconds was implemented (see Section II-B4), and all frames preceding this start time were excluded.

Starting Time	Accuracy
0,0 sec	54 %
0,7 sec	60 %

TABLE III: posture results with initial frames filtering.

With this adjustment, the accuracy for all gestures increased to 60%. This improvement indicated that frame *filtering* was beneficial for our model, as it effectively reduced noise and disturbances.

- **Batch Size Selection**

Experiment 6:

Until this point, the neural network's batch size had been relatively small. An experiment was conducted to evaluate the effects of increasing the batch size; however, no significant change in accuracy was observed.

In such instances, it is preferable to utilize a larger batch size, as it contributes to the stabilization and smoothing of the training process.

Batch Size	Accuracy
32	60 %
128	60 %

TABLE IV: posture results with different batch size values.

- **Dropout Layer**

	attention	right	left	stop	yes	shrug	greeting	continue	turnback	no	slowdown	come
attention	0,56	0	0	0	0	0	0,28	0	0	0	0	0
right	0	0,83	0	0	0	0	0	0	0	0	0	0
left	0	0	0,56	0	0	0	0	0	0	0	0,28	0
stop	0	0	0	0,42	0	0	0	0	0,21	0,21	0	0
yes	0	0	0	0	0	0	0,28	0	0	0,28	0	0,28
shrug	0	0	0	0,17	0	0,17	0	0	0	0	0,33	0,17
greeting	0	0	0	0	0	0	0,56	0,14	0	0	0	0,14
continue	0	0	0	0	0,42	0	0	0,42	0	0	0	0
turnback	0	0	0	0	0	0	0,12	0	0,48	0	0,12	0,12
no	0	0	0	0	0,14	0	0,07	0,07	0	0,49	0	0,07
slowdown	0	0	0	0	0	0,42	0	0	0	0	0	0,42
come	0	0	0	0	0	0	0	0,19	0	0,09	0,37	0,19

TABLE I: Confusion matrix for All Gestures (*X: Prediction, Y: Reality*).

Experiment 7:

After initial attempts proved unsuccessful, more significant modifications were implemented. Specifically, a new Dropout layer with a scale factor of 0.2 was introduced to mitigate overfitting.

Unfortunately, this adjustment led to a substantial decrease in model accuracy, which dropped to 20%. Consequently, this modification was reverted.

Dropout Scale Factor	Accuracy
0	60 %
0,2	20 %

TABLE V: POSTURE results against a default Dropout Layer (0,2 *scale factor*).

Experiment 8:

Theoretically, incorporating a Dropout layer is expected to improve the algorithm's performance. Thus, it may be beneficial to adjust and reduce the scale factor initially used to achieve better results.

A further test was conducted with a Dropout layer using a lower scale factor of 0.05. While this adjustment led to a slight increase in accuracy compared to the previous trial, the improvement was minimal.

Dropout Scale Factor	Accuracy
0	60 %
0,05	25 %
0,2	20 %

TABLE VI: POSTURE results against different Dropout factors.

- **Class Imbalance**

Experiment 9:

After removing the Dropout layer and going back to the experiment number 6, the class imbalance strategy seen in Section II-B5 was tested. Sadly, the accuracy kept steady at 60%.

After removing the Dropout layer and reverting to the setup from experiment number 6, the class imbalance strategy described in Section II-B5 was implemented. Unfortunately, the accuracy remained unchanged at 60%.

B. Model Results

Following the extensive experimentation with the proposed model, a new line of investigation was initiated to understand why the model's accuracy remained at 60%. The primary issue identified was related to the dataset used. Subsequent in-depth analysis revealed the following:

- Not all subjects have been recorded with the same camera, and some of the videos have different number of frames per second (FPS) so the time difference between each graph in the temporal sequence is not constant.
- The length of the different gestures is not constant either, and while some gestures are done in less than a second, others might require a few more.
- Some of ours gestures like, *no*, *yes*, *come* or *slowdown*, have quite a dependency on the hands and finger positioning, but the landmarks extraction method used in this work does not return many nodes per hand.

With all this in mind and despite of not being satisfied with this accuracy, we had to settle.

	Accuracy
Final Model (All)	60%
Dynamic	62 %
Static	82 %

TABLE VII: POSTURE final accuracy results.

C. State-of-the-Art Results

The model proposed in this paper is a modification of Adaptive Graph Attention Networks designed specifically for our dataset (see Section II-B).

Had we employed a dataset that, while less natural than ours, was perfectly labeled and structured, we likely would have achieved better results. One such dataset

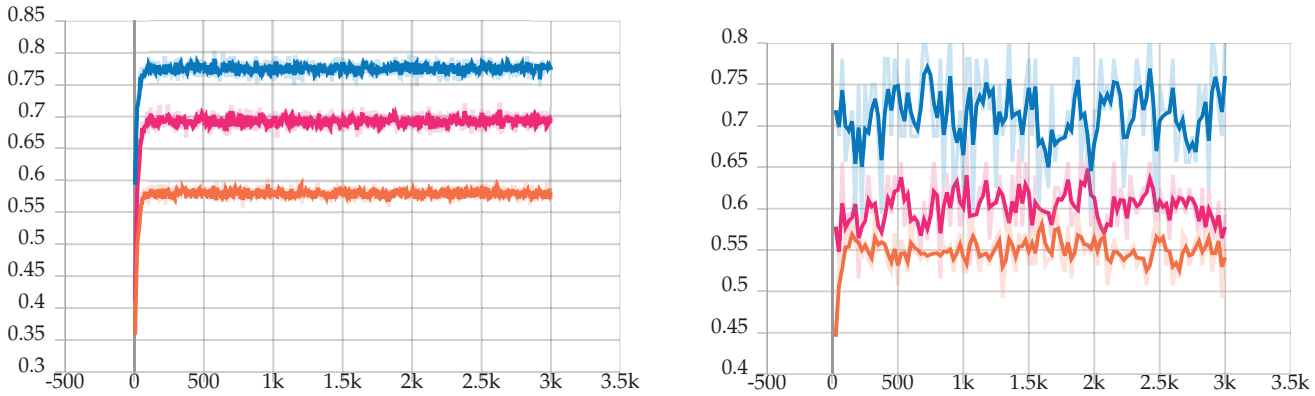

Fig. 5: Train and Test accuracies over epochs for experiments 4 (*blue*), 5 (*pink*) and 9 (*orange*).

is NTU [11] & [12], which has demonstrated accuracy exceeding 90% [9].

This suggests that the primary issue with our current model lies with the quality of our dataset. We attempted to evaluate our model using the NTU dataset but encountered several incompatibilities: (*i*) our nodes include an additional feature (*visibility*); and, (*ii*) the number and arrangement of nodes in the skeleton structure differ (we have 15 nodes, whereas NTU has 25).

These discrepancies necessitated changes to the model's layer configuration, hindering our ability to compare results and proceed with fine-tuning effectively.

IV. Conclusions

The main project objective was to research about Body Gestures recognition for Human Robot Interaction and propose a solution for gesture classification using DL models. Specifically, the proposed model uses the main key concepts of the state of the art solution with a few modifications that have allowed using a custom body gestures dataset.

As a feature extractor, MediaPipe Pose has demonstrated impressive performance in detecting joints and extracting landmarks. However, its precision diminishes when applied to hand detection and still requires further refinement. The extractor's limitation of only four joints per hand, which do not correspond to individual fingers, has posed challenges for gestures requiring precise finger movement differentiation, such as *yes*.

Testing has revealed that gesture recognition performs significantly better with static gestures compared to dynamic ones. This suggests that the temporal component of continuous gestures presents a substantial challenge.

Given that the underlying model has shown remarkable results, it is evident that the primary issue lies with our dataset, which includes variations in gesture length, camera configurations, and resolution.

Despite these challenges, our experiments have enabled us to improve accuracy across all gestures to 60%.

Additionally, the POSTURE model has been trained and tested with real-life scenarios involving continuous gestures, utilizing a batch approach to enhance gesture classification.

References

[1] M. Dalmasso, A. Garrell, P. Jiménez, and A. Sanfeliu, "Human-robot collaborative navigation search using social reward sources," in *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 2*. Springer, 2020, pp. 84–95.

[2] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and Vision Computing*, vol. 25, pp. 1875–1884, 12 2007.

[3] A. Garrell, M. Villamizar, F. Moreno-Noguer, and A. Sanfeliu, "Proactive behavior of an autonomous mobile robot for human-assisted learning," *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pp. 107–113, 2013.

[4] E. Repiso, F. Zanlungo, T. Kanda, A. Garrell, and A. Sanfeliu, "People's v-formation and side-by-side model adapted to accompany groups of people by social robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2082–2088.

[5] J. Laplaza, J. J. Oliver, R. Romero, A. Sanfeliu, and A. Garrell, "Body gesture recognition to control a social robot," 6 2022.

[6] Mediapipe, "Pose." [Online]. Available: https://google.github.io/mediapipe/solutions/pose

[7] R. Romero, "Gesture dataset." [Online]. Available: https://github.com/RamonRL/GESTURE-PROJECT/tree/main/dataset/BodyGestureDataset

[8] "Iri - institut de robòtica i informàtica industrial." [Online]. Available: https://www.iri.upc.edu/

[9] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12 018–12 027, 5 2018.

[10] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 7444–7452, 1 2018.

[11] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2684–2701, 5 2019.

[12] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," *Computer Vision Foundation*, 2016.