HRI-Gestures: Gesture Recognition for Human-Robot Interaction

Avgi Kollakidou[1,*][a], Frederik Haarslev[1,*][b], Cagatay Odabasi[2,*][c],
Leon Bodenhagen[1][d] and Norbert Krüger[1][e]
[1]*SDU Robotics, University of Southern Denmark, Campusvej 55, Odense C, Denmark*
[2]*Fraunhofer IPA, Nobelstraße 12, Stuttgart, Germany*

Keywords: Action Recognition, Gesture Recognition, Human-Robot Interaction.

Abstract: Most of people's communication happens through body language and gestures. Gesture recognition in human-robot interaction is an unsolved problem which limits the possible communication between humans and robots in today's applications. Gesture recognition can be considered as the same problem as action recognition which is largely solved by deep learning, however, current publicly available datasets do not contain many classes relevant to human-robot interaction. In order to address the problem, a human-robot interaction gesture dataset is therefore required. In this paper, we introduce HRI-Gestures, which includes 13600 instances of RGB and depth image sequences, and joint position files. A state of the art action recognition network is trained on relevant subsets of the dataset and achieve upwards of 96.9% accuracy. However, as the network is designed for the large-scale NTU RGB+D dataset, subpar performance is achieved on the full HRI-Gestures dataset. Further enhancement of gesture recognition is possible by tailored algorithms or extension of the dataset.

1 INTRODUCTION

With the technological advancements within the field of robotics, mobile robots are becoming more present in our daily lives and are expected to play an even bigger role in the future (Bodenhagen et al., 2019). Improvements in sensor technology and vision algorithms, especially deep learning, have widened the market for mobile robots, as they can be deployed in more use-cases. Deep learning has shown great potential for tasks such as object detection, pose estimation, object tracking, and action recognition.

In recent years, part of the focus has shifted from improving the accuracy on public datasets, to making the algorithms efficient enough to run on mobile robots. This, combined with improvements in edge hardware, has enabled robots to use, e.g., online object detection for navigation (Chatterjee et al., 2020). While object detection works robustly in unconstrained environments, enabling robots to interact with objects, interaction with humans is still a challenge. A reason for this is that, while objects are inherently static, humans behave dynamically and their actions are hard to predict.

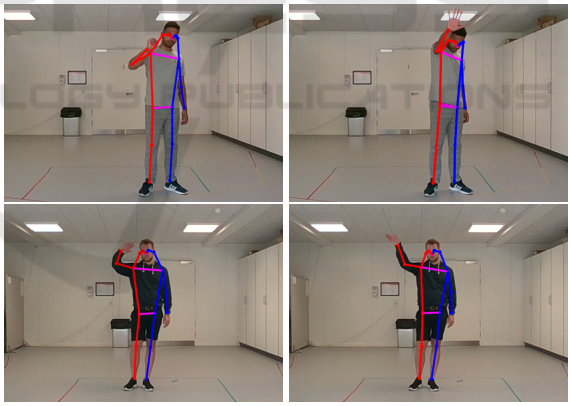

Figure 1: Examples of non-verbal cues used by humans and their detected poses. Up: *Stop*; down: *Get Attention*.

One important facet of HRI is understanding intentions. It is common for mobile robots to signal their intention when navigating, e.g., through lights (Palinko et al., 2020). Additionally, the establishment of mutual gaze via animated eyes is used (Krüger et al., 2021). It is human nature to look into each others eyes when communicating, and thus the intention is instantly recognizable.

[a] https://orcid.org/0000-0002-0648-4478
[b] https://orcid.org/0000-0003-2882-0142
[c] https://orcid.org/0000-0001-5241-2419
[d] https://orcid.org/0000-0002-8083-0770
[e] https://orcid.org/0000-0002-3931-116X
[*]Equal contribution between the authors

Kollakidou, A., Haarslev, F., Odabasi, C., Bodenhagen, L. and Krüger, N.
HRI-Gestures: Gesture Recognition for Human-Robot Interaction.
DOI: 10.5220/0010871200003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 559-566
ISBN: 978-989-758-555-5; ISSN: 2184-4321

An important problem lies with the robots' abilities to interpret humans' intentions. One clear way for a person to indicate their intention to a robot is by expressing it verbally. Speech recognition has improved significantly in the recent years as evident by the emergence of personal assistants like Alexa and Siri. Research has also been conducted on the use of recent advancements in speech recognition in verbal commands for robots (Tada et al., 2020). However, speech only accounts for 30% of communication (Hull, 2016). The rest is non-verbal communication cues, mainly body language and gestures (Figure 1). Hence, if robots are to be accepted as a part of society, humans should be able to communicate with them, possibly as they would with other humans. Accordingly, the ability to perceive gestures is important in facilitating satisfactory HRI.

Gesture recognition entails perceiving how the body moves and determining the meaning of that movement. The problem can be split into three well-defined subproblems: human pose estimation, tracking and action recognition. Human pose estimation and object tracking are heavily researched topics and are largely solved. Juel. et al. (2020) describe a system for detection and tracking of objects and human poses, designed for use on mobile robots. This system is used for human pose detection and tracking throughout this paper.

After detection and tracking of the human pose, the last step in gesture recognition is interpreting the movement of the poses. This is referred to, in literature, as action recognition and is also a well studied topic. Multiple large scale action recognition datasets are currently publicly available (Kay et al., 2017; Liu et al., 2020a). Besides being used as benchmarks for action recognition algorithms, they can be used to train algorithms for detection of various activities in our daily lives, such as *brushing teeth*, *reading*, *drawing*, and *making pizza*. However, they only contain few classes which are relevant for non-verbal communication, including *nodding*, *shaking head*, *thumb up*, *thumb down*, and *pointing to something*. While these classes can be used to express agreement/disagreement or to draw attention to something, they do not provide an expressive non-verbal vocabulary and thus they are not sufficient for the problem of gesture recognition in HRI.

Therefore, in order to facilitate HRI through non-verbal communication recognition, a gesture dataset is required. Such a dataset enables the training of action recognition algorithms, which allow robots to perceive the intentions of humans. For an action recognition algorithm to work with the human pose estimations from an online detector and tracker, it needs to be trained on similar data. The dataset should therefore contain the same pose labels as the output of the human pose estimator used on the robot as well as classes relevant in a HRI context. Such classes include gestures for getting the robot's attention, making it follow you, or making it stop (Figure 6). As such a dataset is, to the best of our knowledge, not available, the creation of one is necessary.

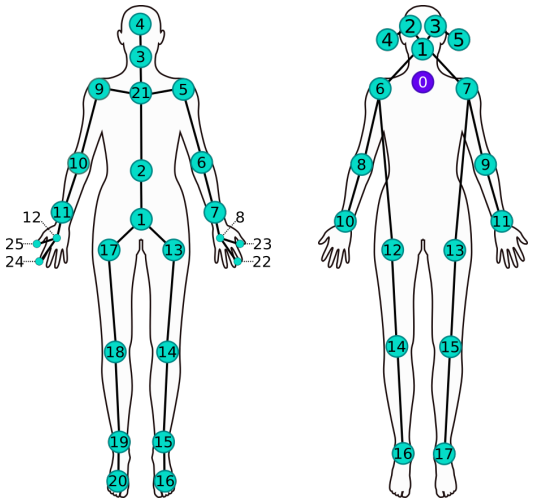

Figure 2: The skeleton models used in the (a) NTU RGB+D and (b) COCO datasets.

In this paper, the HRI-Gestures dataset[1] is presented. 4 RGB-D sensors are used to record 17 subjects performing 15 interactive and 5 passive actions. The interactive actions are gestures directed towards a mobile robot, whereas the passive actions are human behaviors which a mobile robot might encounter when navigating. 3D pose sequences are extracted using the aforementioned human pose estimation and tracking system. An action recognition network is trained on subsets of the dataset created for specific use case scenarios, resulting in gesture recognizers which can be used as is in the relevant use cases.

2 RELATED WORK

In this section the current state of the art action recognition algorithms and datasets are introduced. The NTU RGB+D 120 dataset (Liu et al., 2020a) consists of 120 classes and 114.480 action samples. It contains multiple modalities, including RGB images, depth maps, IR images and 2D and 3D skeletons, all collected using a Kinect v2. Besides containing only few classes for gesture recognition in HRI, another

[1] Available at gitlab.sdu.dk/sdurobotics/HRI-Gestures

problem arises from the use of the Kinect to capture the data. The Kinect skeleton tracking software which is used for extracting the human pose labels has 25 joints in its skeleton model (Figure 2a). However, the COCO keypoint dataset (Lin et al., 2014) which is widely used for training human pose estimation algorithms only contains 17 joints (Figure 2b). A robot would therefore need to have a Kinect v2 in order to obtain sequences which can directly be used for action recognitiontha trained on NTU RGB+D 120.

The Kinetics 400 dataset (Kay et al., 2017) is another large scale action recognition dataset consisting of 400 classes and 306.245 action samples. The original dataset only contains RGB videos, however, the Kinetics Skeleton 400 dataset has been created from it by using OpenPose (Cao et al., 2019) on the videos. This results in 2D pose sequences with the same 17 joints as the output of other popular human pose estimators (Zhou et al., 2019; McNally et al., 2020), as well as an 18th point in the center of the torso (joint "0" in Figure 2b). While these poses can be used for training action recognition algorithms which use the same skeletons as what can commonly be detected on a mobile robot, the lack of depth information poses a problem. Some research (Liu et al., 2020b) shows that skeleton-based action recognition algorithms perform better when using 2D skeletons as input instead of 3D. However, when deploying the algorithm on mobile robots the problem of egomotion arises. This is better handled in 3D, as the poses can be transformed to a static frame resulting in the 3D coordinates not changing when the robot moves, thus eliminating the problem of egomotion.

The dataset is the first part of the problem of gesture recognition. The other part is the action recognition algorithm. Many different models for skeleton-based action recognition have been proposed (Liu et al., 2017; Thakkar and Narayanan, 2018; Song et al., 2020, 2021), and most recent ones are based on Spatial-Temporal Graph Convolutional Networks (ST-GCN) as proposed by Yan et al. (2018). Graph convolutions generalizes the common convolutional layer, as they behave similarly but are not confined to operating on a grid like structure. Instead they can be used on any connected graph structure. ST-GCNs treat skeleton pose sequences as graphs where joints are connected spatially as seen on Figure 2, but also temporally to the same joints at the previous time step.

Song et al. (2021) introduce the Richly Activated Graph Convolutional Network (RA-GCN). It uses multiple ST-GCN streams in a hierarchy where subsequent streams learn discriminative features from inactivated joints from the previous stream. Features from all joints are thereby learned, making the network robust to occlusion and jittering. This is ideal in a mobile robotics context, as detections tend to be noisy when captured online.

Gesture recognition for HRI, therefore, seems possible with the framework presented in this work. Training a state of the art action recognition algorithm such as RA-GCN on a gesture dataset which contains relevant classes for HRI and is created using the same skeleton sequence modalities as what is available to the mobile robot when operating in real-time.

3 GESTURE RECOGNITION

In order to recognize the gestures of people in the vicinity of a robot, it needs to detect how the people move their bodies and then infer a semantic meaning from that movement. With the popularization of using CNNs in computer vision, many previously hard vision problems have become solvable as long as enough labeled data specific to that problem is available. With the data available, the task lies in designing a network which is able to learn from the available data and generalize for unseen data instances. Given the large amount of labeled human pose data which is publicly available, human pose estimators generalize enough to be used reliably on robots in unconstrained environments. However, the current focus in action recognition research is not HRI, meaning that labeled data relevant to gesture recognition for HRI is not publicly available.

While the publicly available action recognition datasets do not contain relevant data for HRI, they have still driven research in action recognition leading to newer and better algorithms (Liu et al., 2017; Thakkar and Narayanan, 2018; Song et al., 2020, 2021). While the gesture recognition for HRI is different than video analysis, the task is nevertheless about deriving semantic meaning from sequences of human poses, meaning that the already existing action recognition algorithms should be transferable to this new domain of gesture recognition for HRI, once a suitable dataset is available.

3.1 Gesture Dataset

In order to address the problem of gesture recognition on a mobile robot, the HRI-Gestures dataset is collected. As the goal of HRI-Gestures is for the robot to detect non-verbal communication during HRI, 15 interactive classes are chosen (Figure 6a-o) where a person attempts to convey information or instructions to the robot: *Stop, Go right, Go left, Come here, Follow me, Go away, Agree, Disagree, Go there, Get at-*

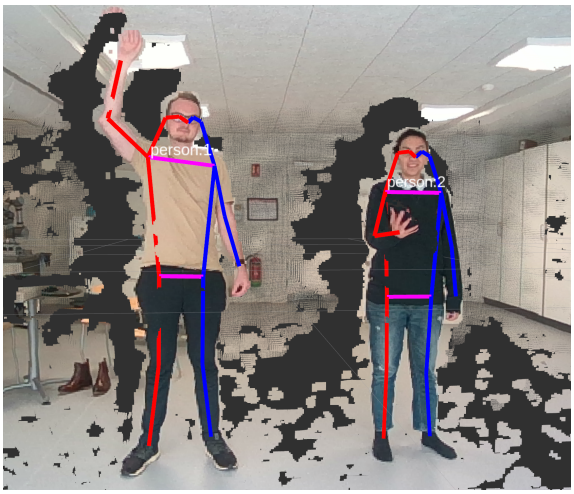

Figure 3: Visualization of the detection and tracking system on people doing the *Get Attention* and *Come Here* action. 2D pose estimations are projected to 3D using the available depth data and the people are assigned a unique ID using the tracker. The 3D poses and ID's are visualized together with the point cloud.

tention, Be quiet, Don't know, Turn around, Take this, Pick up. As a robot is also likely to see people who are not trying to interact with it, 5 passive classes are chosen as well (Figure 6p-t): *Standing still, Being seated, Walking towards, Walking away, Talking on phone*. This results in 20 different classes for the dataset.

The dataset should contain the same modalities the robot can obtain in real-time. Juel. et al. (2020) describe a system for human pose detection and tracking made for mobile robots. The system can be used to obtain 3D pose sequences of people in the field of view of RGB-D cameras on the robot. It detects human poses in 2D, projects them to 3D using the available depth data, and then uses a tracker to sequence the poses belonging to the same person (Figure 3). By recording videos of gesture examples using RGB-D sensors commonly found on mobile robots and then using the pose detection and tracking system to extract pose sequences, the HRI-Gestures dataset was created with input modalities identical to the ones detected real-time on a mobile robot.

Besides using the same input modality as the one available on the robot, the camera positions also plays a role. While some social mobile robots have cameras at heights closer to the eye-level of a person, many mobile robots used today are logistics robots with cameras close to the ground. In order to simulate these differences, four cameras are placed as shown on Figure 4, three RealSense D415 (field of view: 65°x 40°) at different angles close to the ground, and one RealSense D455 (field of view: 87°x 58°), in the head of a social robot. All cameras are calibrated

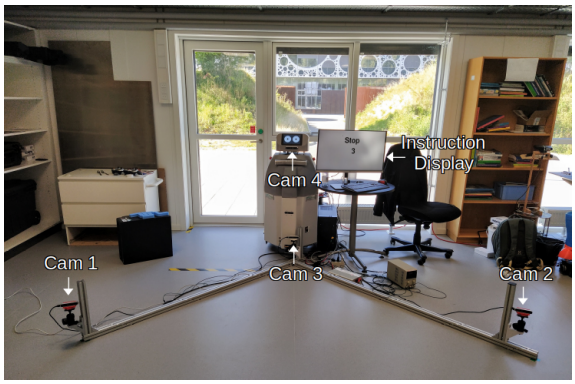

Figure 4: Recording setup. A display instructs the subject which gesture to perform to the robot. This is recorded with four different cameras.

Figure 5: Fields of view from the four cameras (action *Get attention*).

and the calibration parameters are available with the dataset. While recording the interactive classes, the subjects direct their commands towards this robot. The field of view of the cameras can be seen in Figure 5.

Using this setup, 17 adult subjects are recorded performing the 20 actions. Each subject performs the entire set of actions with a randomized order and the process is repeated 10 times. This ensures that the subject is unaware of the sequence of actions to be performed and guaranteed diversity in orientation, placement as well as performance of the actions. The subjects are instructed to keep within a certain area in order to ensure that they stay within the field of view of all cameras. Figure 6 shows each action class being performed by the subjects.

For each recording, the RGB and depth images are saved. Afterwards, these are run through the pose detection and tracking system, resulting in a separate pose sequence for each camera, i.e., four for each repetition. While the cameras record the same performance, the different angles ensure that the extracted pose sequences are not identical. The subjects are not instructed in the nature of their gestures, only the message they should convey. Through this, gesture varia-

(a) Stop (b) Go right (c) Go left (d) Come here

(e) Follow me (f) Go away (g) Agree (h) Disagree

(i) Go there (j) Get attention (k) Be quiet (l) Don’t know

(m) Turn around (n) Take this (o) Pick up (p) Standing still

(q) Being seated (r) Walking towards (s) Walking away (t) Talk on the phone

Figure 6: Actions included in HRI-Gestures. (a) to (o) show the interactive classes, while (p) to (t) show the passive ones.

Figure 7: Difference in gesture performance (action *Pick Up*) between subjects.

tions for each action class can be observed (Figure 7).

The dataset includes the joint position files as well as the raw RGB and depth images. A drawing of the recording setup including distances between cameras and the recording along with camera calibration files are available. The availability of the RGB and depth images ensures that new post-processing methods, e.g. different human pose detection algorithms, are possible in the future.

3.2 Gesture Learning

In order to obtain results on HRI-Gestures, RA-GCN (Song et al., 2019) was trained on the dataset. This model is especially suited for operation on a mobile robot, as it is designed to handle occlusions which commonly occur in robotics applications. This robustness to occlusions is due to the network's multi-stream design, as it is composed of several streams of ST-GCN (Yan et al., 2018).The pose sequences are first input to a ST-GCN stream and the activated joints, i.e., those which contribute the most to the output, are recorded. The activated joints are then masked in the original sequence and become input to a second ST-GCN stream in the 2-stream model. This enables the model to learn rich features from all joints and thus making it more robust to occlusions. The network can also be set up with a third stream which takes the similarly masked sequence of the second stream as input, resulting in a 3-stream model.

The full HRI-Gestures dataset is used to train both 2-stream and 3-stream RA-GCN models using either 2D or 3D poses. The dataset is also split into subsets containing only some of the classes, such as only the interactive or the passive ones, or by merging all the interactive classes and all the passive ones, creating a binary classification problem.

Two splits are introduced for evaluation, Cross-Subject (CS) and Cross-Repetition (CR). In CS, 14 subjects are used for training and 3 subjects are used for evaluation. This evaluates the generalization of the model to different individuals. In CR, repetitions odd numbered repetitions of each subject is used for training while even numbered repetitions are used for evaluation. This enables evaluation of generalization on different instances of the same individual. CR is also an indication of the results achievable in CS if abundant data is available, as action performance variance will be covered.

Table 1: Cross-subject (CS) and cross-repetition (CR) accuracy on full dataset using either 2 or 3 stream RA-GCN model with 2D or 3D keypoints.

Model	d	CS	CR
2s RA-GCN	2D	66.6%	82.3%
3s RA-GCN	2D	67.5%	83.3%
2s RA-GCN	3D	69.0%	83.8%
3s RA-GCN	3D	70.0%	84.9%

4 RESULTS

In this section, results from training the RA-GCN network on the collected dataset are presented. As mentioned before, RA-GCN is a multi stream model, allowing subsequent streams to focus on joint locations which were not in focus on previous streams. 2- and 3-stream models are trained on the CS and CR splits, using either 2D or 3D joints as input modalities. The resulting validation accuracies can be seen in Table 1. The first thing to notice is that CR accuracy is considerably higher than CS. This is because that even though each subject did not perform the actions in the same way through all repetitions, the relative variance in the actions for the same subject is smaller than the variance of the actions between subjects. This shows that the trained models do not fully generalize to new subjects, evident as well by the training accuracies reaching above 99 % in most cases, which indicates that the model overfits the data due to the large difference in training and validation accuracy. Since the RA-GCN network was designed to operate on a much larger dataset such as NTU, the need for a new recognition model, adapted for a smaller-scale dataset becomes apparent. Alternatively, the problem could be solved by collecting additional data.

When comparing 2D against the 3D counterparts, 3D delivers slightly better performance and as 3D is also better suited for use in mobile robotic applications due to the aforementioned problem of egomotion, all the following experiments are conducted using the 3D modality. The 3-stream models for both modalities, achieve marginally better results than their 2-stream counterparts. However, since the 2-stream model is computationally lighter than the 3-stream model, it is better suited for mobile robotic applica-

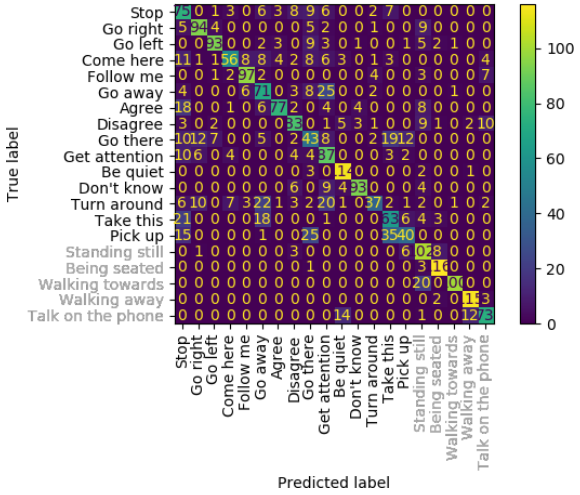

Figure 8: Confusion matrix of 2s RA-GCN with 3D keypoints on CS validation set. Black are interactive and gray are passive classes.

tions and thus is chosen for all further experiments.

In order to further analyze the performance of the model, a confusion matrix is constructed from the results of the 2-stream 3D model trained on the CS split (Figure 8). The confusion matrix shows how many validation examples of each class, which were classified as which classes. Each row corresponds to examples from a specific class, which are classified as corresponding to the column. A correctly classified example is counted in the diagonal, and thus most examples should lie here if the network performs well. The passive classes perform well, as can be seen from the lighter colors attributed to the lower end of the diagonal. *Go there* or *Turn around* on the other hand are troublesome as they are classified wrongly more often than not. This can be due to the action reenactment by the subjects, where both classes were performed with pointing a finger towards a direction, which could also be seen in other classes, e.g. *Go right, Take this*.

In most applications, robots operate within a certain context which limits the amount of relevant actions and gestures. Several cases were selected, e.g., recognizing whether a person is attempting to interact with the robot or not, and separate networks were trained for those contexts by dividing the dataset in subsets. Two subsets were created using the interactive and passive classes separately. Independent networks are trained on the subsets. The results can be seen on Table 2 with the equivalent subset name. The performance on CR is, as expected, better and it is also clear that passive classes are more easily recognized than interactive ones. This means that a network trained on the "passive" subset could be used in scenarios where a robot is navigating in a human occupied environment with no intention of interaction.

Table 2: Subsets of classes used to partition the dataset and train individual models.

Subset	CS	CR
Interactive	65.3%	81.0%
Passive	93.6%	97.1%
Binary	95.9%	98.2%
Go	68.1%	84.1%
Agreement	49.6%	52.7%

A "binary" subset was created by merging the interactive classes into one class and the passive classes into another. The binary subset shows whether the subject is attempting to interact or not with the robot and could be used in such a use case in mobile robots, to clarify human's intentions towards the robot.

"Go" describes all classes with the intention of indicating a direction or goal to the robot (*Go there, Go right, Go Left, Go away*). The subset shows considerably better results than the individual actions in the original model but still similar to the overall CS accuracy of the entire dataset, which is inadequate.

"Agreement" includes only the *Agree* and *Disagree* classes, evaluating the distinction between the two. In contrast with the rest, this subset learning is poor. As it was observed during recordings a popular depiction of the actions consisted of thumbs up and thumbs down gestures, and since the skeleton joints used, do not include the thumbs or fingers, these could not be learned. This shows that the selected joint skeleton is not suitable at this point for these classes.

5 CONCLUSIONS

In this paper, the problem of gesture recognition for human-robot interaction is addressed and analyzed. Gestures are a crucial component of communication in human-robot interaction, and thus it is something which robots should be able to detect in order to improve their HRI capabilities.

In order to solve the problem of gesture recognition, it was identified that a proper public gesture dataset is missing. This paper has presented a methodology for creating such a dataset which can be used for training algorithms usable on mobile robots in unconstrained environments.

The methodology has then been used to create the HRI-Gestures dataset. Subsets of the dataset can be used for gesture recognition in various HRI contexts, e.g., by training a network to distinguish between in-

teractive and passive classes (reaching 95.9%) in order to determine whether someone is trying to interact with the robot or not.

The results show that, with our approach, gesture recognition with high classification rates is possible for important subtasks in HRI. On the full classification issue of 20 classes, our method achieves 70%. A different joint constellation could improve results on classes that rely on finger joints, which are not included in the dataset.

Further enhancement of gesture recognition is possible. Extending the dataset or creating algorithms which achieve higher accuracy on the full HRI-Gestures dataset could be considered.

ACKNOWLEDGEMENTS

This research was supported by the HanDiRob project, funded by the European Fund for regional development, and by the DIREC project, funded by Innovation Fund Denmark.

REFERENCES

Bodenhagen, L., Suvei, S.-D., Juel, W. K., Brander, E., and Krüger, N. (2019). Robot technology for future welfare: meeting upcoming societal challenges – an outlook with offset in the development in scandinavia. *Health and Technology*, 9:197–218.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chatterjee, S., Zunjani, F. H., and Nandi, G. C. (2020). Real-time object detection and recognition on low-compute humanoid robots using deep learning. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 202–208.

Hull, R. (2016). The art of nonverbal communication in practice. *The Hearing Journal*.

Juel., W., Haarslev., F., Krüger., N., and Bodenhagen., L. (2020). An integrated object detection and tracking framework for mobile robots. In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, A., Suleyman, M., and Zisserman, A. (2017). The kinetics human action video dataset. *ArXiv*, abs/1705.06950.

Krüger, N., Fischer, K., Manoonpong, P., Palinko, O., Bodenhagen, L., Baumann, T., Kjærum, J., Rano, I., Naik, L., Juel, W., Haarslev, F., Ignasov, J., Marchetti, E., Langedijk, R., Kollakidou, A., Camillus Jeppesen, K., Heidtmann, C., and Dalgaard, L. (2021). The smooth-robot: A modular, interactive service robot. *Frontiers in Robotics and AI*, 8.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755.

Liu, H., Tu, J., and Liu, M. (2017). Two-stream 3d convolutional neural network for skeleton-based action recognition. *ArXiv*, abs/1904.07850.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. (2020a). Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701.

Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020b). Disentangling and unifying graph convolutions for skeleton-based action recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 140–149.

McNally, W. J., Vats, K., Wong, A., and McPhee, J. (2020). Evopose2d: Pushing the boundaries of 2d human pose estimation using neuroevolution. *ArXiv*, abs/2011.08446.

Palinko, O., Ramírez, E., Juel, W., Krüger, N., and Bodenhagen, L. (2020). Intention indication for human aware robot navigation. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: HUCAPP*.

Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2020). Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia (ACMMM)*.

Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2021). Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1915–1925.

Song, Y.-F., Zhang, Z., and Wang, L. (2019). Richly activated graph convolutional network for action recognition with incomplete skeletons. In *International Conference on Image Processing (ICIP)*. IEEE.

Tada, Y., Hagiwara, Y., Tanaka, H., and Taniguchi, T. (2020). Robust understanding of robot-directed speech commands using sequence to sequence with noise injection. *Frontiers in Robotics and AI*, 6:144.

Thakkar, K. C. and Narayanan, P. (2018). Part-based graph convolutional network for action recognition. *ArXiv*, abs/1809.04983.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *ArXiv*, abs/1801.07455.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *ArXiv*, abs/1904.07850.