Predictive intention recognition using deep learning for collaborative assembly

Khansa Rekik*, Nishant Gajjar†, Grimaldo Silva‡ and Rainer Müller§

ZeMA gGmbH
Saarbrücken, Germany,
Email: *k.rekik@zema.de, †nishantkgajjar@outlook.com, §rainer.mueller@zema.de

SENAI CIMATEC and UNEB,
Salvador, Brazil
Email: ‡jose.jgrimaldo@gmail.com

***Abstract*—Achieving seamless human-robot collaboration in tasks with uncertainty requires anticipating human intentions. In order to overcome the limitations of classical techniques for inferring human intentions, which often struggle due to delayed articulation and lack of knowledge about spatio-temporal dependencies, this study proposes an approach centered around real-time prediction of human intention using Long-Short Term Memories (LSTMs). To achieve this goal, the human hands and assembly product components were detected in each frame, using a novel dataset, the former is then used as the primary input to our LSTM model. Finally, we demonstrate and validate the effectiveness of this framework in a real industrial assembly scenario, where a robotic agent utilizes the predicted intention to efficiently assist the human in successfully completing the assembly of a product. The results indicate, with high confidence, a response time about three times faster using our approach, on average, compared to waiting for the object detection module to recognize that a component was removed from a work bin.**

***Index Terms*—hand segmentation, intention recognition, deep learning, long short term memory**

I. Introduction

Solving a problem that involves cooperation requires, in tasks with at least some level of uncertainty, a certain synchronization between teammates. This would enable agents involved in the problem-solving to anticipate each other's needs, avoiding potential bottlenecks in production. Whenever human and robot are involved in such tasks, some means of preempting their actions through direct or indirect communication can be lost e.g. recognition of body language, hand signals, or speech. Autonomous robots could compensate for this lost communication by anticipating their human teammates' plans based on their recent actions. Thus, this work tackles human-robot collaboration with the hypothesis that estimating human intention based on their recent actions can improve collaboration between human-robot teams.

Oftentimes in industrial scenarios, tasks have to dynamically adapt to unforeseen events such as faulty components or missing equipment. Our work focuses on one particular case where such events happen: an industrial micro-controller housing assembly. The housing construction can start from multiple distinct initial human actions, however, each action requires a specific reaction from the robot to proceed to the next step of the assembly.

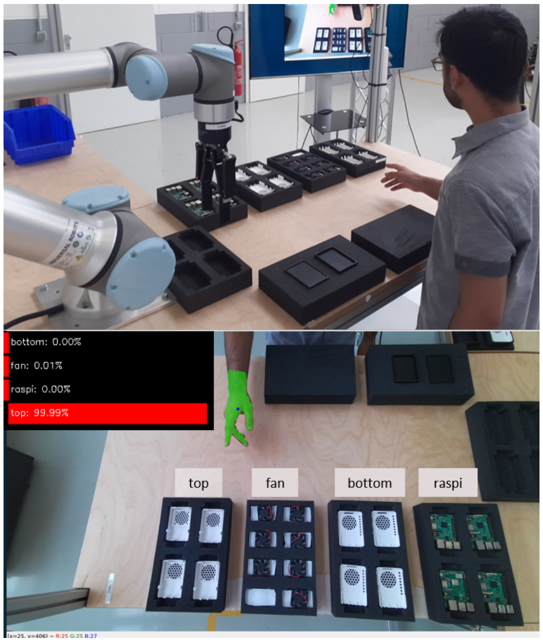


Fig. 1: Collaborative task of micro-controller housing assembly (upper image). In the bottom image, the robot correctly estimates user intentions and is able to preemptively choose the correct action.

Consider a scenario, such as shown in Fig. 1, where a person decides to attach the upper part of the housing to the micro-controller. Based on that, the robot should start attaching the lower part of the micro-controller housing to the fan. Thus, the casing can be put together in the next step. Conversely, if the person decides to start from the lower part, the robot should adapt accordingly.

The aforementioned term "adaptation" hides an underlying problem – it could cause a robot to wait until a person finishes their action in order to react. Our work tackles this problem through the estimation of human intention based on recent actions. For that, two contributions are described in this work. First, two datasets were created, one composed of about a thousand images focusing on labeled data for hand segmentation models, while the other one is composed of hundreds of short videos with hand motion. From the two datasets created, the one related to hand segmentation has been made publicly available[1]. Our last contribution is a method for human intention estimation based on a Long-Short Term Memory (LSTM) model [1].

For analysis of our results, a number of live tests are performed using the aforementioned datasets. Furthermore, more live tests using the complete system are executed to assess its usability in the real world along with its run-time speed.

II. Literature Review

The application of collaborative robots has demonstrated significant effectiveness in diverse tasks like assembly, construction, and inspection. To establish successful collaboration between humans and robots, it is vital for robots to possess the capability of identifying and predicting human targets and intentions. This involves monitoring and predicting human actions within the framework of the overall task plan, facilitating a unique form of collaboration where robots predominantly rely on implicit cues rather than explicit instructions from their human counterparts to guide their actions.

Several notable studies have delved into the field of Human-Robot Collaboration (HRC). In [2] and [3], comprehensive insights into the cutting-edge technologies and methods employed in this area are offered. [4] propose a framework that utilizes human body gestures to command robots in manual assembly lines within the automotive industry. In a comparable manner, [5] explores the utilization of Human-Robot interaction in urban search and rescue scenarios, including how robots' data assists humans spread across different locations.

In [6], a framework was proposed for HRC using a combination of Spatial-Temporal Graph Convolutional Networks and Long Short-Term Memory Networks, specifically for handheld object identification based on YOLOv3. While Graph Convolutional Networks are effective in understanding the relationships between nearby data points, their computational cost, memory consumption, and training time increase as the number of points grows. In contrast, [7] utilized Convolutional Neural Networks (CNNs) with Long-Short Term Memory to accurately predict human motion in a computer disassembly task, overcoming the computation overhead of VGGNet and achieving faster evaluation.

Perception plays a pivotal role in facilitating Human-Robot interaction, as it enables the robot to gain awareness of its environment and the precise nature of the product to be handled [8]. For instance, in assembly tasks, the deployment of well-suited vision sensors empowers the robot to effectively detect objects, obstacles, and components, thereby acquiring a comprehensive understanding of the ongoing assembly process. Advanced techniques, such as object detection and instance segmentation, have demonstrated promising outcomes in enhancing perceptual capabilities, thereby contributing to the overall effectiveness of collaborative tasks.

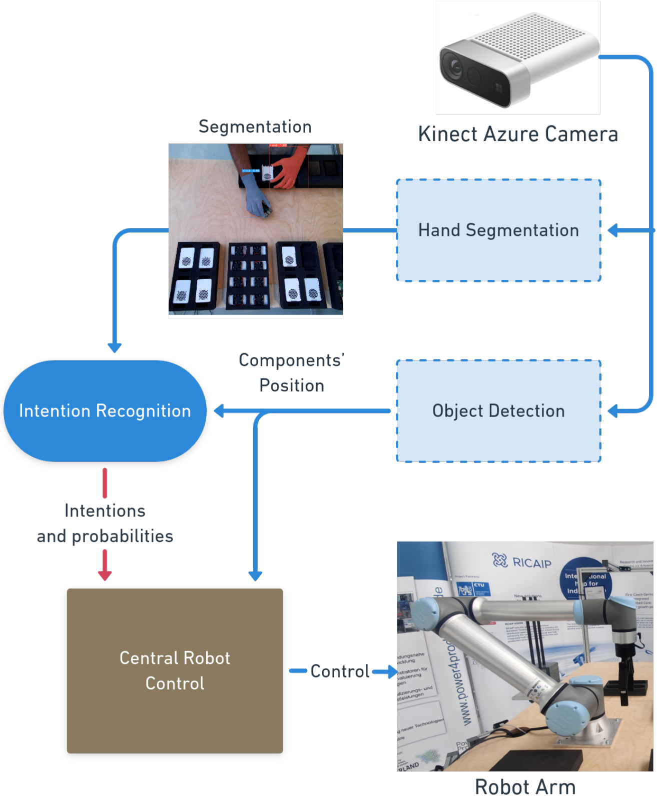
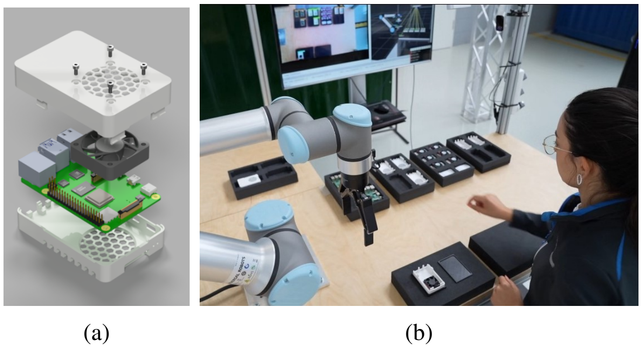


Fig. 2: Architecture of our approach. Dashed lines indicate modules that have been previously developed. The main output of this work is in red color, that is, a mapping of the human partner's possible intentions and an associated probability to each intention.

Numerous research studies have made significant contributions to the domains of object detection and hand recognition. For instance, in [9] a real-time object detection technique using single-shot detectors was introduced, offering applicability across diverse devices and environments. Moreover, [10] presented a comprehensive method that integrates depth features, object contours, and hierarchical segmentation, leading to precise object detection and segmentation. [11] employed a cascaded Mask R-CNN approach to effectively detect essential objects in assembly scenarios. Similarly, [12] leveraged a pre-trained YOLOv5 model to enhance the assembly process in human-robot collaboration systems.

Regarding hand detection, [13] devised a fully-labeled approach for detecting hands in dynamic Ego-Centric videos, albeit with limitations in real-time performance. [14] introduced a rapid hand detection and tracking method utilizing a single depth camera, demonstrating robust performance and fast inference times; however, this method is contingent upon the availability of depth data. [15] proposed a real-time hand

[1] Dataset can be found in github.com/jgrimaldo/intentionrecog/



(a) (b)

Fig. 3: The use-case components and setup: (a) Exploded view of the Product made of the complete case, the Raspberry Pi, and the fan (b) the collaborative assembly cell layout.

recognition framework with a particular focus on hand gesture recognition using CNNs. [16] utilized Mobile-Net and SSD architectures to achieve efficient and accurate hand detection, employing feature maps of varying resolutions to accommodate different hand sizes. Nevertheless, their approach, based on single-shot detectors, prioritizes speed over performance, which may introduce the possibility of false positives.

III. Extended Modules for Task

This section first defines the scope of the task being tackled. Afterward, to better accomplish this task, the extensions made to the previously developed modules are presented. The extensions made to these modules, enabling their integration with our work, are also described. The main modules required for intention estimation are shown in the architecture overview in Fig. 2.

A. Task Description

The use case investigated in this work consists of an assembly task (shown in Fig. 3b) where an HRC-capable robot is to assist a human worker in completing the task in a human-like fashion.

The product to be assembled consists of four main components - a 3D printed case composed of a top part and a bottom part, a Raspberry Pi micro-controller, and a 40 mm DC fan that is intended to be bolted to the top part for cooling purposes (as shown in Fig. 3a). The components are sorted in different bins in a way that facilitates their handling using the robot gripper. Moreover, two depth cameras are used with a field of view covering the assembly area. The first camera tracks human hand motions to infer intentions, while the second camera is for object detection.

The assembly of the product can be performed in different orders, e.g. starting from the top part followed by the fan, bolting, bottom part then Raspberry Pi, closing the case then palletizing the finished product or the other way around, it is important for the task sharing approach to accommodate the dynamics of the task in a way that does not encumber the worker and spares them unnecessary and long waiting times. In other words, it is to be avoided that the robot agent starts its action after the human finished theirs, as that would lead to a waiting phase on the latter's side. It should be noted that certain tasks can only be performed by human workers, such as bolting the fan and closing the case, as they require flexible hand control. On the other hand, there are tasks that can be performed by both humans and autonomous robots interchangeably.

B. Real-time Hand segmentation

Numerous cutting-edge methodologies have emerged for the purpose of object detection and instance segmentation. However, a majority of these techniques tend to prioritize either segmentation performance or run-time speed. Our task requires real-time execution with acceptable detection performance. Preferred techniques for such requirements include SSD [17] and YOLO [18].

In an attempt to correctly balance these trade-offs, a technique named *You Only Look At Coefficients* (YOLACT) is used. The main concepts governing the choice and configuration of the YOLACT model are described in our previous work [19].

Although the results in [19] showed promise, its smaller dataset limited the ability of YOLACT of generalizing to distinct users and scenarios. For that, we further extended its training dataset to a little over 1000 positive image samples with different people and even more variation in hand positions. This dataset is made publicly available in this work[1].

As shown in Fig. 2, the outcome of the hand segmentation is fed to the intention recognition module. It consists of hand keypoint coordinates as well as its depth value with respect to the camera in the form of a time-series 3-dimensional tensor $[X_1, Y_1, Z_1] \cdots [X_t, Y_t, Z_t]$. It is restricted to a single keypoint per hand to avoid noise and null points during occlusion and overlapping hands. Nevertheless, null detections can still occur in the single keypoint approach yet they can be recovered by means of interpolation.

C. Assembly Object Detection

As can be seen in Fig. 3b, each work cell is composed of multiple objects that are either work tools or materials for the micro-controller assembly along with partially and fully assembled components.

Detecting and tracking these objects, as they move along the work cell, was done in [19]. For that, the first step is preprocessing the image using Gaussian filters in order to smooth out noise. Afterward, the image is binarized using adaptive thresholding with a value that is calculated based on the Gaussian mean within each region. Finally, each grid region containing a distinct component, such as fans or micro-controllers, is obtained through hierarchical extraction of contours from which corner points are extracted. These corner points are used to perspective transform the detected grid into a flattened image that can then be indexed by grid position.

These grid positions are a part of the required input for our intention recognition.

IV. Intention estimation from recent actions

For a smooth and human-like collaboration, the robot agent needs to infer subsequent goals of the human at each step by predicting their intentions, then perform helping assembly steps. This section deals with a deep-learning-based approach used to infer human intentions based on the history of their hand motions.

In our aforementioned previous work [19], a basic method for predicting hand heading was introduced. It uses the Euclidean distance between the hand keypoint and the center of each bin on 2D images to detect where the hand is closest to and starting from a specific threshold determines the heading i.e. the intended object. Due to its simplicity, this method fails to capture temporal and spatial relations between consecutive frames which is crucial for a successful prediction.

To address these issues, we not only incorporate depth information but also adopt an approach that can extract spatial representations of specific features in sequential data. This enables us to accurately predict probable human intentions.

A. Deep learning-based Model

After segmenting each instance of the human hand in the environment, as introduced in Sec. III-B, we feed the output to the deep-learning model specifically a Long-Short Term Memory model.

Introduced by [1] in 1997, LSTMs are a type of Recurrent Neural Networks (RNN) with feedback connections that enable capturing and retaining long-term dependencies in sequential data.

An LSTM network is composed of several units, also called cells, each unit is made of an input gate (i_t), an output gate (O_t), a forget gate (f_t), and a cell state (C_t).

For the purposes of this work, a PyTorch-based model is developed, operating as follows:

- Model input: the series of 3-dimensional tensor $[X_1, Y_1, Z_1] \cdots [X_t, Y_t, Z_t]$ representing the hand keypoint coordinates.
- Model output: classification based on the anticipated destination of the hand, specifically into the categories: $[Bin-Bottom, Bin-Raspi, Bin-Top, Bin-Fan]$.
- Network: It is fundamentally composed of multiple recurrent hidden layers and a final output layer. In order to capture more intricate temporal patterns across various time steps, multiple LSTM layers are stacked. Thus facilitating spatial distribution of the model parameters and leading to faster convergence ([20]). Additionally, a dropout layer is incorporated into the network to prevent overfitting on the training data.

B. An Intention Dataset

To train the aforementioned LSTM model, a custom dataset has been created. It consists of over 200 videos of different hands moving to grasp one of the objects from all four bins. The data is divided into training, testing, and validation sets. The data is captured by means of an Azure Kinect camera [21] using both the depth and RGB channels and processed at first using the YOLACT model of Sec.III-B. To ensure data diversity, videos have been recorded with different users with different hand features. Each recording starts when the hand starts moving towards a bin and ends once an object is grasped resulting in an average length of 2.5s, about 75 frames, per video, depending on the pace of the user.

Each frame is then passed to the Hand Segmentation module to determine the (X, Y) coordinates of the hand's keypoint. Afterward, these keypoints are projected on the depth map corresponding to the frame in question in order to determine the Z coordinate. In the subsequent step, the processed data is input into the LSTM model in the form $[X_{yolact}, Y_{yolact}, Z_{DepthMap}]$.

Note that various interpolation techniques are employed to handle missing data or undetected hand movements in some frames. More specifically, if the hand cannot be accurately identified in the environment, its value would be updated as *NaN*, then over batches of size 75, each value is scanned, and missing values are addressed by applying linear interpolation in forward and then backward direction. This procedure has proven to be efficient even for the first and last elements of the batch as it handles all the missing values as equally spaced and finds the best fit with respect to the previous value.

C. Discussion

The number of valid worker configurations per work cell is limited to one. Deviation from this prescribed configuration, for example, a worker standing at a different edge of the table, may result in inaccurate LSTM predictions. Specifically, while workers may have the flexibility to approach cells from various sides, only the single most optimal configuration for the given task setup is considered valid for the purposes of our system.

Although this limitation reduces some flexibility, similar requirements are common even in conventional industrial cell layouts. As such, addressing this issue is beyond the scope of this study.

V. Evaluation

After briefly describing our experimental setup, the evaluation section of this study focuses on two aspects of our solution. The first aspect involves determining the best set of hyperparameters, such as learning rates and choice optimizers, for our LSTM model. The second aspect shifts the focus to real-world evaluation, where we assess the practical benefits of the LSTM model for intention recognition.

A. Experimental Setup

This work is experimentally evaluated on a system with an Intel core™ i7-8700 CPU with a total of 6 cores and 12 threads running a base frequency of 3.20 GHz, 32 GB RAM, and an NVIDIA Geforce GTX 1080 GPU with 8 GB VRAM, then deployed to use-case introduced in Sec. III-A.

The demonstrator, shown in Fig. 3b, consists of a multi-cell setting for pre-assembly, collaborative assembly, and palletizing stations. A collaborative robot arm, UR10e [22], equipped with a gripper is mounted in the middle of the multi-cell so

Optimizer	ADAM	NADAM	RMSPROP	SGD
Train Accuracy	**0.8345**	0.8257	0.8264	0.5784
Train Loss	0.3974	**0.3899**	0.4128	1.3451
Val. Accuracy	**0.8449**	0.8321	0.8322	0.5541
Val. Loss	0.3685	0.3651	**0.3477**	1.3984

TABLE I: Loss and Accuracy for optimizers on LSTM

LR	1E-6	1E-5	1E-4	5E-4	5E-4
Train Accuracy	0.8111	**0.8843**	0.8736	0.8184	0.2687
Train Loss	0.6120	**0.2818**	0.2850	0.4176	1.3783
Val. Accuracy	0.8232	0.8877	**0.8940**	0.8793	0.3032
Val. Loss	0.5776	**0.2789**	0.2901	0.3803	1.3114

TABLE II: Loss and Accuracy for different learning rates on LSTM

that it is able to reach all mentioned stations, two 3D Azure Kinect cameras are mounted above the assembly station to monitor the environment and the human worker.

B. Preliminary Results

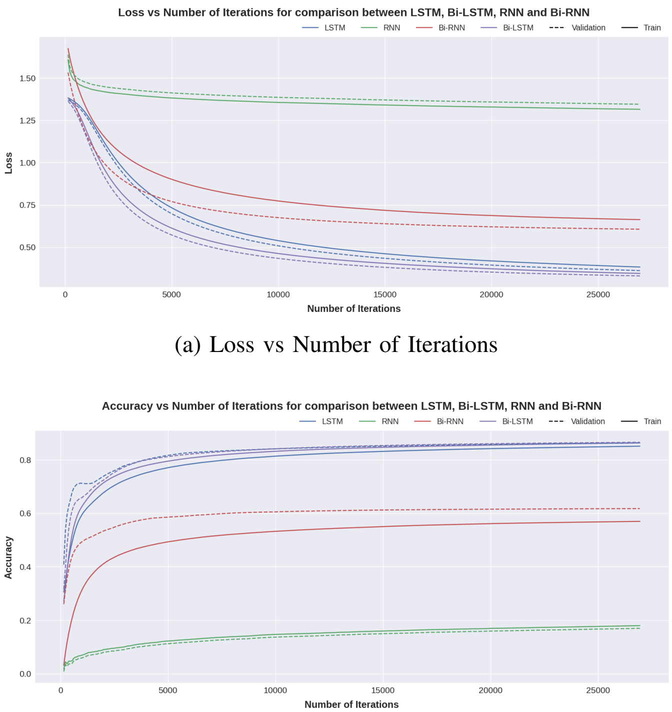
1) Numerical Performance Assessment: The impact of various parameters on the model is investigated to assess the performance of our LSTM-based intention recognition model. These parameters include various learning rates, the configuration of stacked and unstacked LSTM layers, the number of hidden layer units in the LSTM network, different layer depths, and different optimizers. Additionally, a comparison with other recurrent neural network (RNN) variants is conducted, namely RNNs, bidirectional RNNs (Bi-RNNs), and bidirectional LSTMs (Bi-LSTMs).

First, the performance using multiple optimizers is tested, keeping the remaining hyperparameters constant. Based on the results shown in Table I, we opt for ADAM as it yields faster and more stable convergence with accuracy near 85%.

In addition, the effect of various learning rates is investigated. As shown in Table II, it has been noted that a model trained with a learning rate of *1e-4* was the most stable and achieved an accuracy of around 89%. Estimation failures were, among other reasons, caused by partial occlusion of the person's hand by the robot's arm or hesitation of the person on the correct action to perform, leading to backtracking or similar actions.

Furthermore, stacking multiple LSTMs proved fruitful, as they outperformed a single-unstacked LSTM model. More specifically, with *3* LSTMs an average accuracy of over 89% was achieved. Finally, varying the number of the hidden layers, shows that as their number of units increases, the model is better able to generalize. A *512* hidden layer units-model has the best performance with an accuracy near 90%, beyond which it reaches a plateau and some instability is observed.

Finally, an optimal LSTM model was compared against an optimal RNN model as well as both their bidirectional variants. As shown in Fig. 4 LSTM and Bi-LSTM outperformed RNN by quite a large margin. Similarly, LSTMs and Bi-LSTMs achieved accuracy in the range of 89% versus 67% and 20% for Bi-RNNs and RNNs respectively. Although Bi-LSTMs performance was comparable to the unidirectional one, the former took longer to train, thus, we opted for the latter.



(a) Loss vs Number of Iterations

(b) Accuracy vs Number of Iterations

Fig. 4: Comparing LSTM, RNN, Bi-RNN and Bi-LSTM in terms of Loss and Accurany.

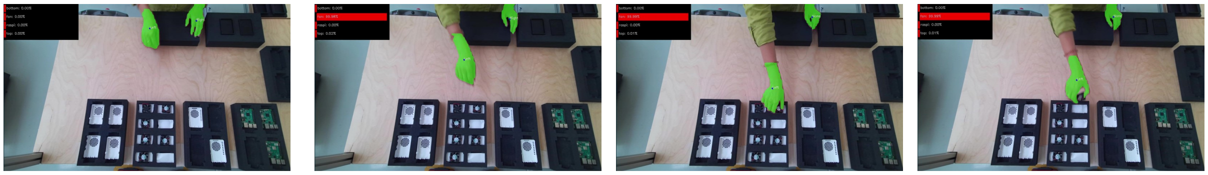
2) Assessment on Demonstrator : The presented modules i.e. Intention Estimation and Object Detection were integrated into the aforementioned demonstrator and orchestrated using Robot Operating System (ROS). To evaluate the intention estimation, we ran multiple tests and noted for each: the time required by our model to submit a prediction along with the time until a hand reaches a bin.

The system has been tested on different manipulation speeds representing different human workers' profiles. Slow manipulation is usually observed among inexperienced workers and when the target object is distant from the worker.

Our approach is compared to an object detection-based one where the action performed by the human is inferred once a missing object is detected in one of the bins (see Fig. 5d).

For predictions, our method relied on acting based on the highest probabilities within a specific window, which smoothed out short-term fluctuations, focusing on stable trends. The window size was determined empirically and linked to a probability matrix. Experiments showed that adjusting the window size has minimal impact on predictions, with larger windows offering only marginal benefits. Moreover, our threshold is set high enough to minimize uncertainty.

As exemplified in Fig. 5, on average, the system could predict the human intention with a high confidence level in about $1/3$ of the action time. The mean time for human action



(a) Human hands detected at $t = 0$s

(b) Intention predicted at $t = 0.30$s

(c) Human reached object at $t = 0.86$s

(d) One less Object detected inside bin at $t = 2.20$s

Fig. 5: Human hand motion sequence for grasping a fan. In this example, it took 0.3s to predict the intention, 0.86s for the full action whereas the object detection-based approach took 2.2s to infer the human action.

is 1.24 seconds while our prediction requires 0.43 seconds. In average, our approach executes 0.81 seconds faster, with a variance of 0.047 and a standard deviation of 0.218.

The results mentioned above enable the robot to start acting before the object is grasped and minimizing waiting times for the worker. Certain factors such as the speed of the hand, its orientation, and lighting conditions, can cause detection failures or fluctuations over a few frames as the hand moves. During these fluctuations, no intention can be predicted. However, since the prediction is continuously updated, no false predictions have been observed during the experiments. In cases where the human intentionally tries to deceive the model, the prediction confidence levels decrease.

VI. Conclusion

This study introduces a novel approach to enhance human-robot collaboration by anticipating human intentions. It addresses limitations of traditional methods used to infer human goals and actions, which often struggle with delayed articulation and often neglect spatio-temporal dependencies.

A point not addressed was the impact of external factors like noise or lighting conditions and scenarios with multiple humans, which would affect accuracy of intention prediction. Future work should focus on enhancing model robustness by incorporating a wider range of situations and a task planning module to overcome the manually established relationship between human intention and robot action.

Acknowledgment

The work is part of the RICAIP project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 857306.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] A. Bauer, D. Wollherr, and M. Buss, "Human-robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.

[3] B. Chandrasekaran and J. M. Conrad, "Human-robot collaboration: A survey," in *SoutheastCon 2015*. IEEE, 2015, pp. 1–8.

[4] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, and G. Chryssolouris, "On a human-robot collaboration in an assembly cell," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 580–589, 2017.

[5] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.

[6] C. Liu, X. Li, Q. Li, Y. Xue, H. Liu, and Y. Gao, "Robot recognizing humans intention and interacting with humans based on a multi-task model combining st-gcn-lstm model and yolo model," *Neurocomputing*, vol. 430, pp. 174–184, 2021.

[7] Z. Liu, Q. Liu, W. Xu, Z. Liu, Z. Zhou, and J. Chen, "Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing," *Procedia CIRP*, vol. 83, pp. 272–278, 2019.

[8] K. Rekik, R. Müller, J. Hoffmann, and M. Vette, "A communication-free human-robot-collaboration approach for aircraft riveting process using ai probabilistic planning," *Int. Journal of Advances and Current Practices in Mobility*, vol. 2, no. 2020-01-0013, pp. 1160–1167, 2020.

[9] A. Kumar, Z. J. Zhang, and H. Lyu, "Object detection in real time based on improved single shot multi-box detector algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–18, 2020.

[10] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European conference on computer vision*. Springer, 2014, pp. 345–360.

[11] J. Lee, S. Lee, S. Back, S. Shin, and K. Lee, "Object detection for understanding assembly instruction using context-aware data augmentation and cascade mask r-cnn," *arXiv preprint arXiv:2101.02509*, 2021.

[12] Y. Y. Liau and K. Ryu, "Status recognition using pre-trained yolov5 for sustainable human-robot collaboration (hrc) system in mold assembly," *Sustainability*, vol. 13, no. 21, p. 12044, 2021.

[13] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2013, pp. 3570–3577.

[14] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3213–3221.

[15] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, "Real-time hand gesture detection and classification using convolutional neural networks," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019, pp. 1–8.

[16] L. Yang, Z. Qi, Z. Liu, H. Liu, M. Ling, L. Shi, and X. Liu, "An embedded implementation of cnn-based hand detection and orientation estimation algorithm," *Machine Vision and Applications*, vol. 30, no. 6, pp. 1071–1082, 2019.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[19] N. Gajjar, K. Rekik, A. Kanso, and R. Müller, "Human intention and workspace recognition for collaborative assembly," *IFAC*, vol. 55, no. 10, pp. 365–370, 2022.

[20] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 103–118.

[21] Microsoft, "Azure kinect sensor sdk," Mar 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/kinect-dk/sensor-sdk-download

[22] Universal Robot UR10e. Accessed March 29, 2023. [Online]. Available: https://www.universal-robots.com/products/ur10-robot/