Intention Recognition with Recurrent Neural Networks for Dynamic Human-Robot Collaboration

Matija Mavsar[1], Miha Deniša[1], Bojan Nemec[1], and Aleš Ude[1,2]

***Abstract*— A new method to recognize the intention of a human worker while performing a collaborative task with a robot is proposed. For this purpose, two recurrent neural network (RNN) architectures capable of predicting the worker's target were developed. The first uses marker-based tracking of hand positions and the second RGB-D videos of human motion. The system was implemented to perform a collaborative assembly task. The results show high intention prediction accuracy for both networks, with accuracy increasing once a larger portion of human motion has been observed, making the proposed method viable for efficient and dynamic human-robot collaboration. Furthermore, we developed a framework that enables online adaptation of robot trajectories based on estimated human intentions.**

I. INTRODUCTION

Human-robot collaboration is moving from laboratories to factory floors, where humans and robots share their workspace to effectively accomplish tasks that may be too complex for robots alone. It is essential in such environments that robot and human workers can dynamically share their tasks, i.e., a human can help a robot to perform a task when appropriate. This trend is driven by the rising number of affordable collaborative robots on the market. For the successful implementation of human-robot collaboration, a number of new software tools and algorithms need to be developed. It is important that they are not tailored only to a specific application but generally applicable. Only in this way can we shorten the deployment effort of new solutions and further increase the viability of collaborative systems in industrial settings. One of the problems that needs an adequate solution is the recognition and anticipation of human worker motion.

Recurrent neural networks (RNNs) are a promising technology for collaborative tasks that require anticipation of an agent's motion since they can take advantage of long short-term memory (LSTM) units to analyze time-dependent processes [1]. This allows for predictions of future states based on the previous ones. For example, RNNs are capable of predicting future body poses based on measurements of past poses [2] and labeling of human actions [3].

In this paper, we propose an end-to-end recurrent neural network design to analyze variable-length motions of a human worker's arm and perform classification into a limited set of possible intentions. An additional objective was to develop cost-effective solutions without sacrificing performance, i.e., using off-the-shelf RGB-D cameras. The predictions are used to control and adjust the motion of a collaborative robot sharing workspace with a human worker. The proposed RNN can output the predicted worker intention every time a new sensor measurement has been processed. Based on these results, the robot can change its motion and perform a different operation. To enable smooth transitions between different motions, the robot trajectories are encoded using dynamic movement primitives (DMPs) [4]. We developed and compared two neural networks based on data from two different sensors: a marker-based tracking system and an RGB-D camera. In the first case, the input to the RNN was the operator's hand trajectory, and in the second case, an RGB-D video of the task. The output is in both cases the worker's intention. With this comparison, we wanted to verify that comparable results could be obtained using a more general and affordable solution.

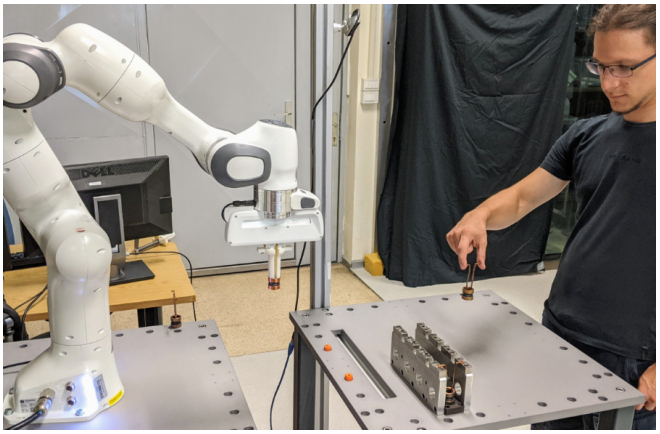

Fig. 1. Collaborative task used for evaluation. The human worker inserts a copper ring into one of the four designated slots, while a robot does the same from the opposite side. To enable dynamic and simultaneous performance of the task, our proposed method can predict the worker's intention and adjust the robot's motion to enable effective use of the shared workspace.

The viability of the proposed approach was shown in a real-life industrial setting (see Fig. 1). In this setup, both the robot and the human collaborated to insert a copper ring into the model for casting car parts. Predictions of human intention were used online to adjust the robot's motion, thus preventing possible conflicts and increasing the efficiency of the task.

II. RELATED WORK

Human-robot collaboration (HRC) has been increasingly studied over the past decade due to growing requirements

[1]Humanoid and Cognitive Robotics Lab., Dept. of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia {matija.mavsar, miha.denisa, bojan.nemec, ales.ude}@ijs.si

[2]Faculty of Electrical Engineering, University of Ljubljana, Slovenia

for service robot applications in home and industrial environments [5], where humans and robots form a system to accomplish a task. Research is focusing on increasing task performance, enabling effective robot learning through physical interaction [6], [7], as well as ensuring task fluency [8]. The development of interfaces for improved perception of the collaborative environment aims to improve such cooperation. Towards this end, the estimation of human motion from video sources has been investigated since many years [9].

Intention recognition is a vital part of HRC, enabling the robot to recognize and anticipate human actions. In [10] a neural network was used for the active leading of robot end-effector by estimating human intention based on current force, position, and velocity measurements. Methods for RNN-based activity recognition and description have also been developed [3], where a description label is predicted from input RGB-D videos. Similar approaches use human skeleton motions as input data to predict either future poses [11], [12] or action probability distribution [13]. In order to recognize whether a handover should take place, authors in [14] have employed support vector machines to distinguish between handover and non-handover motions based on the giver's kinematic behaviors. Another framework learns motion models using probabilistic principal component analysis for motion onset detection and intent estimation [15].

Dynamic Movement Primitives (DMPs) [4] have been used in the context of HRC to provide smooth robot trajectories, since they can smoothly pull the robot towards the desired goal position while following the desired trajectory, even when switching from one desired motion to another. This can be especially useful in dynamic and uncertain tasks that often arise in HRC. To estimate the location of object handover based on hand position measurements, Widmann et al. [16] employed an extended Kalman filter for prediction of DMP parameters to encode the possible giver trajectories. An alternative method for trajectory representation, probabilistic movement primitives [17], has been used to predict the observed motion trajectories in order to coordinate the motions during the object handover task. DMPs have also been employed in image-to-motion translation tasks [18]. A method for optimized backpropagation when training neural networks to predict DMPs has been developed [19].

The benefit of our method is that it doesn't require specialized body motion trackers and is capable of processing variable-length RGB-D videos. The approach most similar to ours is presented in [3] but uses constant-length videos. In addition, in this paper we explain how to apply the output of the neural network in the context of human-robot collaboration.

III. INTENTION RECOGNITION METHOD

The proposed method aims to predict the human worker's intention in an assembly task to achieve a dynamic human-robot collaboration in a shared workspace. We assume that a robot and a human perform a limited set of repetitive tasks. For successful cooperation, we need a system that can recognize the intended version of the task the human is performing and adjust the robot motion accordingly if needed, i.e., if a conflict would arise in the shared workspace. We applied two different neural network architectures that can predict the intention of the human worker; the first, called OptiNet, uses hand position measurements as input, while the second, denoted as HandNet, makes predictions directly from RGB-D videos of the worker's motion. The latter approach allows for the use of accessible off-the-shelf cameras and does not require a motion tracker system to obtain hand positions.

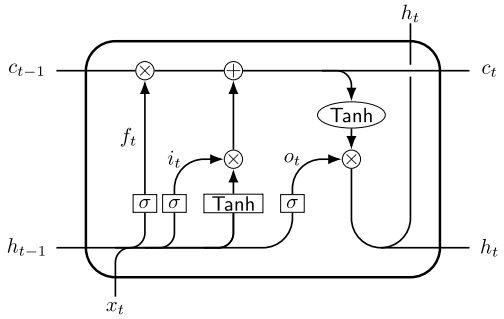

Fig. 2. Structure of an LSTM cell. The cell inputs are the hidden state h_{t-1} and cell state c_{t-1} from the previous time step, as well as the current input data x_t. New cell state c_t and new hidden state h_t are defined by the inputs and the trainable parameters of the LSTM unit, with i_t, f_t and o_t representing outputs of input, forget and output gates.

A. LSTM recurrent neural networks

As previously explained, recurrent neural networks are especially useful for the task of intention recognition from a sequence of input data (in our case, hand positions and RGB-D frames) due to their structure, consisting of memory units that allow storing information dependant on inputs from previous states. However, classic RNNs can suffer from vanishing or exploding gradients [20], which instigated the development of LSTM networks [1]. LSTM memory units are composed of a cell, an input gate, an output gate and a forget gate (Fig. 2). The cell memorizes values over an arbitrary number of time intervals, where the three gates regulate the flow of information into and out of the cell. During training, errors flow backwards through a number of virtual layers unfolded through time, and the LSTM can thus learn tasks that require memories of events from several time steps earlier [21].

B. Proposed neural network architectures

The structures of the proposed networks are shown in Fig. 3. OptiNet consists of fully connected, dropout, LSTM, and softmax layers, while HandNet has additional input convolutional layers combined with group normalization, non-linear and max-pooling layers. The LSTM layers enable processing sequences of input data and make predictions even after a single input sample is processed. This property enables us to recognize the human's intention before the entire motion sequence is available to the network. The

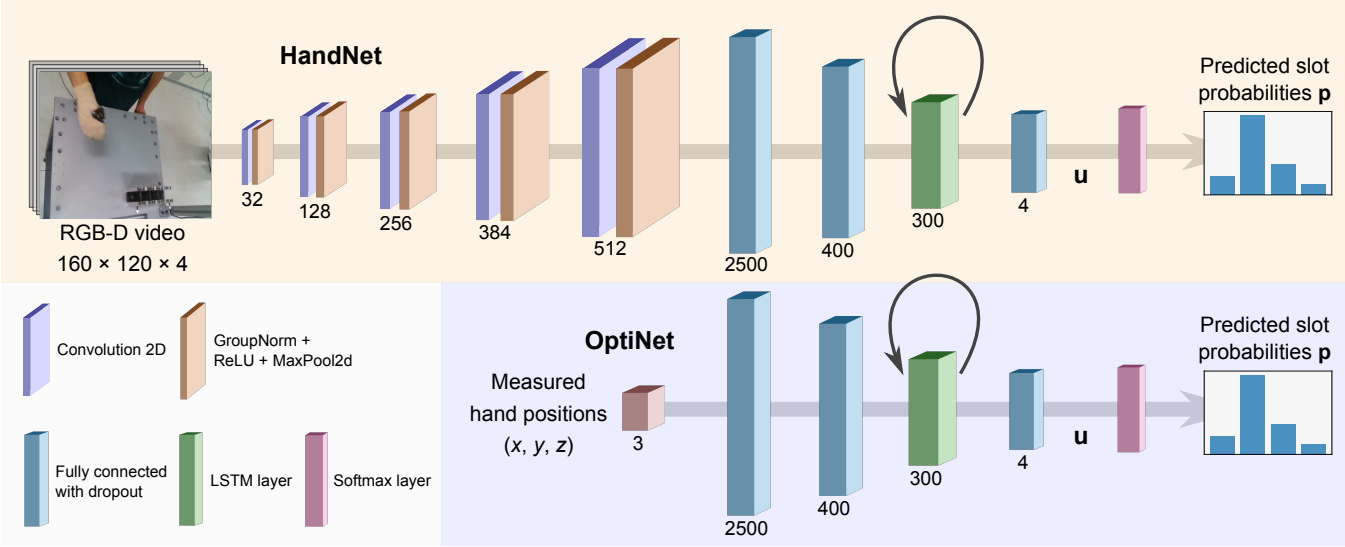

Fig. 3. The proposed recurrent neural network architectures. The structure of both HandNet and OptiNet is similar but they differ in the input layers. HandNet processes RGB-D videos using convolutional layers, while OptiNet infers directly from the measured Cartesian positions of the worker's hand. The outputs $\mathbf{u}$ are sent through a softmax layer to obtain task version probabilities $\mathbf{p}$. For implementation of our use case, we set the input RGB-D frame size to $160 \times 120 \times 4$ and the number of output probabilities – corresponding to the target slots – to 4.

inputs to OptiNet are sequences of Cartesian space hand position measurements $\mathbf{y}(t) \in \mathbb{R}^3$ while the inputs to HandNet are sequences of frames $\mathbf{F}(t) \in \mathbb{R}^{W \times H \times 4}$. Frame $\mathbf{F}$ is an RGB-D image of $W \times H \times 4$ pixels, where W and H are the width and height of input camera frames. Each input sequence is labeled with the ground-truth version of the task $k \in \mathbb{N}$, $k \in \{1, \ldots, m\}$, e.g., target slot for placing of an object, where m is the number of possible targets.

The outputs $\mathbf{u} \in \mathbb{R}^m$ of both architectures are sent through a softmax layer in order to represent a probability distribution $\mathbf{p} \in \mathbb{R}^m$ across the m possible classes. Thanks to LSTM layers, the variations of the task can be predicted after an arbitrary number of input positions or images have passed through the network. The prediction accuracy typically increases as more input samples are obtained during the task execution. The data pairs used for training the OptiNet are thus defined as

$$\mathbf{D}_o = \{\{\mathbf{y}_{ij}\}_{i=1}^{L_j},\ k_j\}_{j=1}^{M}, \tag{1}$$

while the data pairs for training the HandNet are

$$\mathbf{D}_h = \{\{\mathbf{F}_{ij}\}_{i=1}^{L_j},\ k_j\}_{j=1}^{M}, \tag{2}$$

with M being the number of training samples and L_j denoting the length of the j-th input sequence.

C. Cross-entropy loss

To implement the loss function during training of OptiNet and HandNet architectures, cross entropy minimization is employed. The softmax layer ensures that the output values $\mathbf{u} = (u_1, u_2, \ldots, u_m)$ are normalized into a probability distribution $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ over m possible versions of the task:

$$p_i = \frac{e^{u_i}}{\sum_{j=1}^{m} e^{u_j}}, \quad i = 1, \ldots, m. \tag{3}$$

The sum of probabilities in $\mathbf{p}$ thus equals 1, i.e., $\sum_{i=1}^{m} p_i = 1$. Negative log likelihood loss is calculated afterwards. Given a predicted probability distribution $\mathbf{p}_n$ in time step n and a correct target class k, the loss is therefore defined as

$$\mathscr{L}_n(\mathbf{p}_n, k) = -\log(p_{n,k}), \tag{4}$$

where $p_{n,k}$ represents the predicted probability of target class k in time step n.

The losses are calculated for each time step. A weighted sum is used to decrease the significance of early input values (camera frames or pose measurements) and increase the significance of later values. For each input sequence of length L, the total loss is defined as

$$\mathscr{L} = \frac{1}{L} \sum_{n=1}^{L} \gamma_n \mathscr{L}_n, \tag{5}$$

where γ_n represents the weight for the n-th input, computed using a logistic function as

$$\gamma_n = \frac{1}{1 + e^{-5\frac{n-1}{L-1} + 0.5}}. \tag{6}$$

1) Training method: The HandNet and OptiNet architectures were implemented using PyTorch [22] and a NVIDIA GeForce GTX 1080 graphics processing unit. Both architectures were trained using the RMSprop optimization algorithm [23] with a learning rate of 0.0001 and a batch size of 20, where the training was stopped after 40 consecutive epochs of no mean accuracy improvement on the validation set.

D. Dynamic movement primitives

Each task from the set of the robot collaborative tasks is encoded as a DMP. The choice of DMPs makes it possible to provide a smooth transition when switching among different

trajectories. The RNN networks in Fig. 3 give outputs after each new input sample (position measurement or RGB-D image). Thus, the predicted intention of the human worker may change during the collaborative task. Generally, the prediction accuracy is improving as more inputs are available. This requires that the robot is capable of adapting its trajectory while carrying out the assembly task, which can be effectively achieved by utilizing DMPs.

The trajectories of the controlled robot with d degrees of freedom are defined as $\mathbf{y}(t) = [\mathbf{c}(t)^{\mathrm{T}}, \mathbf{q}(t)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{y}(t) \in \mathbb{R}^d$, and consist of Cartesian space positions $\mathbf{c}(t) \in \mathbb{R}^3$ and orientations $\mathbf{q}(t) \in \mathbb{R}^4$. Using DMPs [4], a robot trajectory can be described with the following system of differential equations:

$$\tau \mathbf{z} = \alpha_z(\beta_z(\mathbf{g}-\mathbf{y}) - \mathbf{z}) + \mathrm{diag}(\mathbf{g}-\mathbf{y}_0)\,\mathbf{f}(x), \quad (7)$$

$$\tau \dot{\mathbf{y}} = \mathbf{z}. \quad (8)$$

Here $\mathbf{y}_0 \in \mathbb{R}^d$ represents the initial position of the desired trajectory and $\mathbf{g} \in \mathbb{R}^d$ is the final position or goal of the trajectory. The auxiliary parameter $\mathbf{z} \in \mathbb{R}^d$ is the scaled velocity of motion, obtained with the temporal scaling term τ, and $\mathbf{f}(x) \in \mathbb{R}^d$ denotes a nonlinear forcing term, where $x \in \mathbb{R}$ is the phase defined by

$$\tau \dot{x} = -\alpha_x x. \quad (9)$$

The introduction of the phase removes the direct time dependence of the DMP. The forcing term $\mathbf{f}(x)$ is a linear combination of normalized radial basis functions (RBFs), defined with weights, and can thus be used to approximate an arbitrary function. In this way, the dynamic system can reproduce any smooth motion between the initial and final robot position.

To compute the next position, velocity, and acceleration from the previous ones, the system of equations (7) – (9) must be solved. Parameters α_x, α_z and β_z are usually set to fixed values to ensure critical damping and convergence of $\mathbf{y}$ and $\mathbf{z}$ to a unique attractor point at $\mathbf{y} = \mathbf{g}$, $\mathbf{z} = 0$.

As previously explained, new RNN predictions may require for the robot to switch between different tasks, each described with a unique set of DMP parameters. The smooth switching is provided in the following way. We denote the current DMP integration state as $\mathbf{y}_\mathrm{c}, \mathbf{z}_\mathrm{c}$, the temporal scaling factor of the current trajectory as τ_c, and the temporal scaling factor of the desired new trajectory as τ_n. In order to ensure that the position and velocity of the robot remain smooth during switching, we initialize the next DMP integration states $\mathbf{y}_\mathrm{n}$ and $\mathbf{z}_\mathrm{n}$ as

$$\mathbf{y}_\mathrm{n} = \mathbf{y}_\mathrm{c}, \quad (10)$$

$$\mathbf{z}_\mathrm{n} = \frac{\tau_\mathrm{n}}{\tau_\mathrm{c}} \mathbf{z}_\mathrm{c}. \quad (11)$$

Starting with values (10) – (11), we can continue the integration from current phase x using the DMP parameters of the new trajectory. The initialization values are not guaranteed to lie on the new desired trajectory. However, since DMPs define a control policy, the integration converges to the new desired motion.

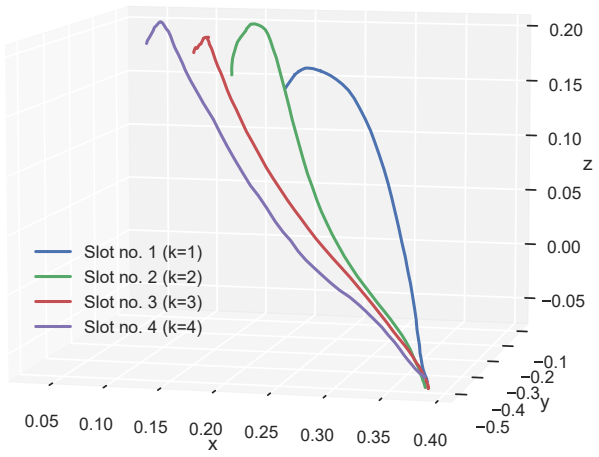

Fig. 4. Example trajectories of hand motion for all four versions of the task, recorded using the OptiTrack motion capture system. Each path belongs to a different task variation, i.e., goal slot label k.

IV. EXPERIMENTS

The aim of the experiments was to validate the OptiNet and HandNet intention recognition architectures. Our goal was to evaluate the accuracy of the prediction of the collaborative task performed (Fig. 1) using both networks. The networks were trained using training and validation sets, detailed in Sections IV-A and IV-B, while their accuracy was calculated on the test dataset.

A. Experimental setup and data acquisition

The experimental setup is presented in Fig. 1. It mimics a real-life industrial scenario from production of car starters. The tasks of the robot and the human were the same, i.e., to insert copper rings into the casting model, and the tasks are executed simultaneously. During collaboration, the human may attempt to insert the ring into the same slot as the robot. Based on the information obtained from hand positions or the video, the robot adapts its work plan to the human operation.

The casting model is composed of four insertion slots, where the robot and the human can access all slots. The challenge is to identify the slot where the human operator intends to place the copper ring as early as possible and modify the robot plan accordingly to prevent possible conflicts in the shared workspace.

During data acquisition the human subject was instructed to place an object from a starting point on one end of the table into one of the four available slots on the other end, while wearing OptiTrack motion capture markers on the hand. The human carried out diverse motions, with an attempt to mimic semi-constrained production environments, where the workers typically execute fluent and non-random motions. Examples of four recorded trajectories, with each leading to a different target slot, are shown in Fig. 4. At the start of the motion, the subject signalled the beginning of video recording and hand pose capturing at a rate of 30 Hz, performed by Intel RealSense Depth Camera D435 and OptiTrack V120:Trio camera system, respectively. When

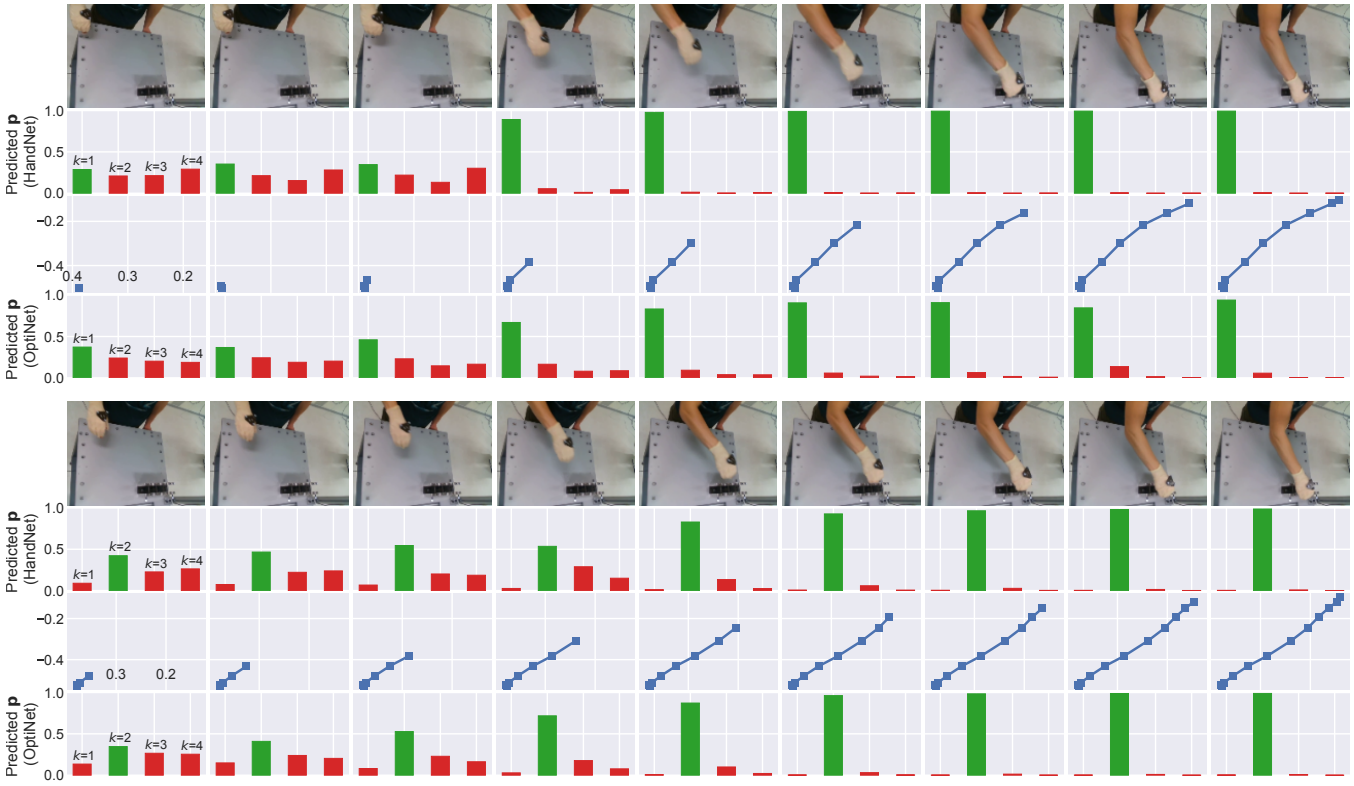

Fig. 5. Example predictions of probability distributions across four goal slots during a task. Each group of four rows belongs to the same motion sample, where the second row of each group are HandNet predictions based on camera frames from first row, and the fourth row are OptiNet predictions based on position measurements in the third row. Green bars represent the probability of the correct goal slot and typically increase as motion progresses. For simplicity, the depth camera frames are not shown, and the trajectories are only shown in the X-Y plane, since Z coordinates are less informative.

the worker reached one of the object slots, the recording was stopped. In this way, a database of 1200 samples was obtained, consisting of RGB-D videos, pose measurements $\mathbf{y}$ and task version labels k in the form of a number from 1 to 4, representing the slot, where the worker placed the object. A subset of 100 motion samples was used as test data, while the rest was used for training and validation and underwent additional processing (see Section IV-B).

B. Data processing

In order to compensate for the low number of video samples in the train/validation database, randomization was implemented to increase the database size. By introducing randomization, we also aimed to improve the networks' capability to successfully generalize to previously unseen data.

The training and validation samples were randomly processed multiple times. Temporal randomization was applied, where a random amount (4 to 16) of RGB-D videos and hand position measurements was extracted. This way, variation of motion lengths was increased. During this process, a larger part of RGB-D videos was transformed with medium randomization (rotations, contrast, saturation and hue changes, noise), while a portion underwent heavy randomization (additional random cropping and resizing, perspective transformations and noise). Validation videos were processed with minimal variations. Test data was temporally subsampled by extracting every 7th sample and no video randomization was applied. During this process, camera frames were resized from the original size of 640×480 pixels to the HandNet input size of 160×120 pixels.

The final number of training, validation and test motion samples was 3200, 200 and 100, respectively.

C. Results

The OptiNet and HandNet networks were evaluated on the test database of 100 motion samples, which were not used during the training phase. The input samples (either sequences of position measurements $\mathbf{y}$ or sequences of RGB-D camera frames $\mathbf{F}$) were passed through the OptiNet and HandNet architectures to obtain the predicted intention of the human worker, i.e., the label of the target slot, where the worker is moving the object.

After each element of the input sample is processed, the networks output a probability distribution across four target slots. With each new position measurement or camera frame, the predicted probabilities are updated, thus allowing online acquisition of the worker's intention as the motion is being carried out. Fig. 5 shows example outputs of both networks for two different motion samples.

To assess and compare the intention recognition accuracy, classification confusion matrices were calculated. Fig. 6

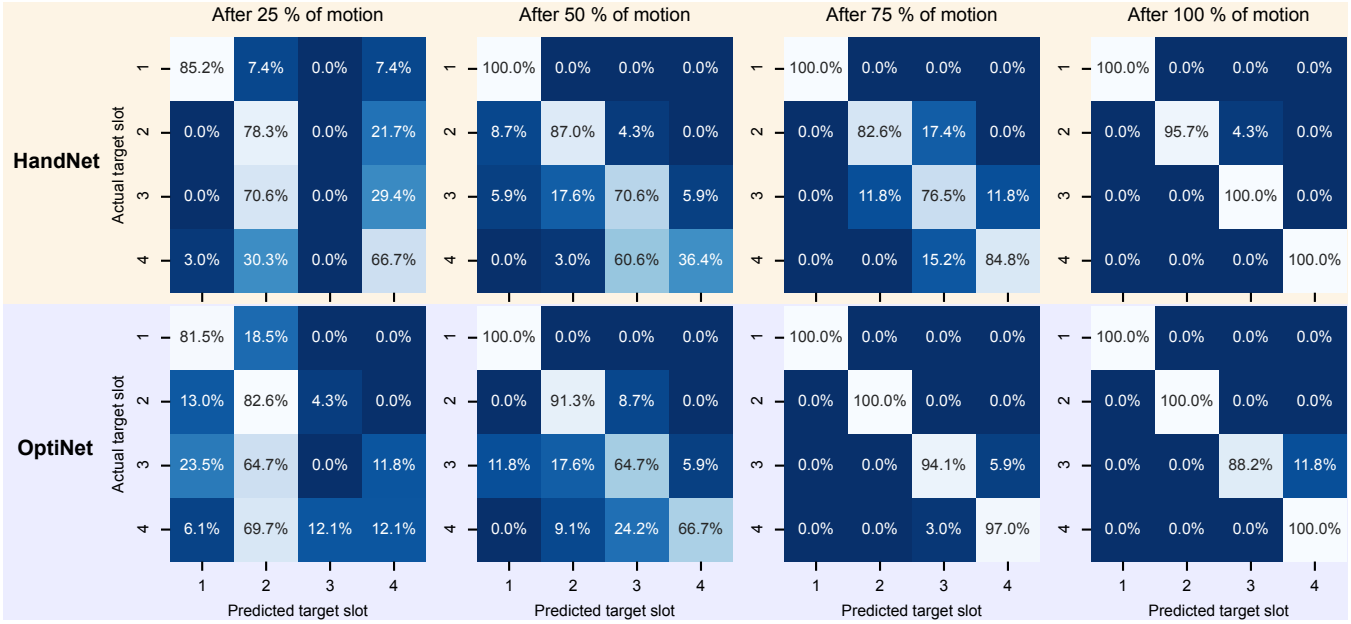

Fig. 6. Confusion classification matrices showing the ratios of correct target slot predictions for the test dataset. Accuracy is shown after processing 25%, 50%, 75% and 100% of observed motion. Value in row $a \in \{1,\dots,4\}$ and column $b \in \{1,\dots,4\}$ denotes the percentages of all samples for the slot a, that were classified as the slot b (the sum of each row is therefore 100%). The matrices show a significant increase in prediction accuracy as a larger part of the motion is processed, with HandNet accuracy being somewhat lower than OptiNet accuracy.

shows confusion matrices for both networks after processing 25%, 50%, 75% and 100% of motions. The values represent the percentages of a certain row slot, that was classified as a corresponding column slot. It is evident that the accuracy is higher when a larger part of the input motion is available to the network. A large portion of samples at 25% of motion are classified as slot $k = 2$, which can be attributed to similar initial parts of trajectories, where the network does not have enough information to determine the goal. OptiNet exhibits slightly better performance overall (above 94% after 75% of motion), with HandNet reaching at least 76% accuracy at 75% of motion, while both networks achieve nearly 100% accuracy after an entire motion. Additionally, the confusion matrices show that the wrongly predicted slots are often the slots that are adjacent to the correct one, except with very early predictions. This can still enable an adequate robot response in certain cases.

A more detailed graph, showing recognition accuracy of the networks in relation to the percentage of the input motion processed, is depicted on Fig. 7. At the start of the motion, the accuracy is lower, since the trajectories to different goal slots are very similar in the initial part. A gradual increase can be observed for both networks as a larger part of the input motion, in the form of Cartesian positions or camera frames, is processed. HandNet predictions show a significantly higher classification accuracy in early predictions, which may indicate that the RGB-D frames contain additional information, such as human body posture and location, and can thus be utilized to infer the target slot from multiple input features, not entirely from hand positions.

The OptiNet performance is evidently slightly higher than

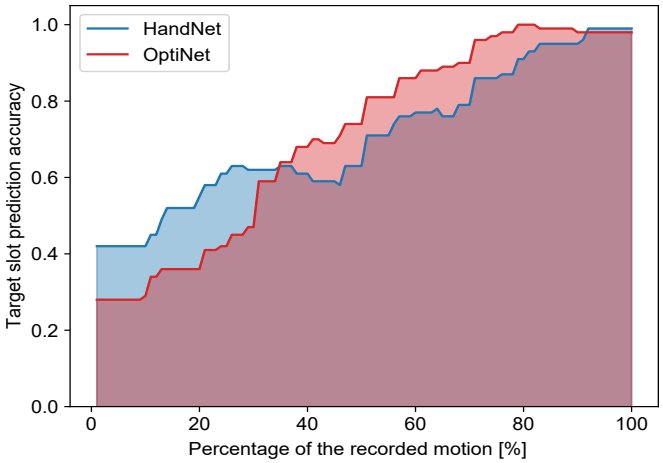

Fig. 7. A graph of recognition accuracy for HandNet and OptiNet, related to the percentage of the input motion that is processed by the networks. An increase of accuracy is apparent as a larger part of motion is available, with the accuracy of both networks being above 60% after half of processed motion and nearly 100% towards the end of motion.

that of HandNet, especially as a large part of motion is processed. Nonetheless, the accuracy of HandNet architecture is still viable for use in HRC tasks to increase the efficiency of cooperation.

D. Implementation of a human-robot collaborative task

The proposed system for online intention recognition and robot control was implemented on a human-robot collaborative task, where a human and a Franka Panda robot worked simultaneously opposite each other to move an object

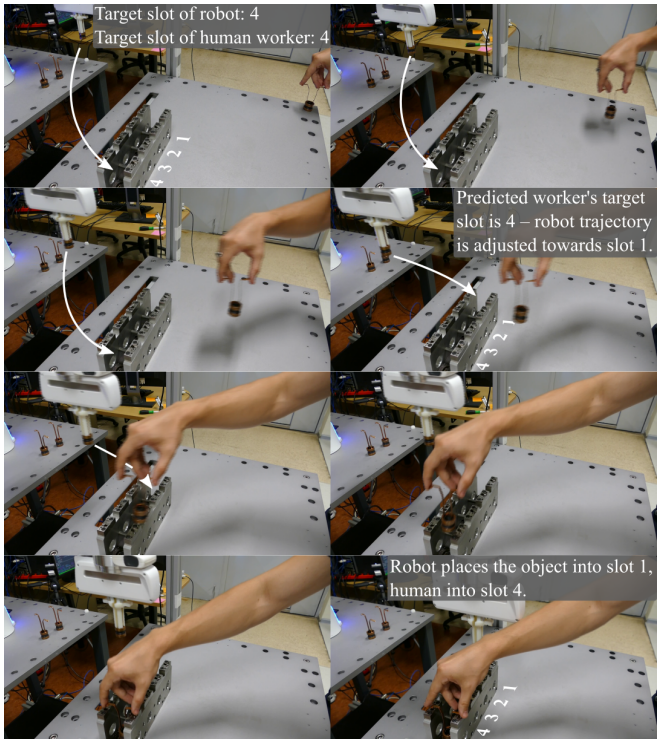

Fig. 8. An example of human-robot cooperation during the implemented experiment. In this example, both human and the robot planned to insert the object into slot $k = 4$. When HandNet recognized the human intention to be the same as the robot's, the trajectory of the robot was adjusted, leading the robot towards the farthest slot $k = 1$.

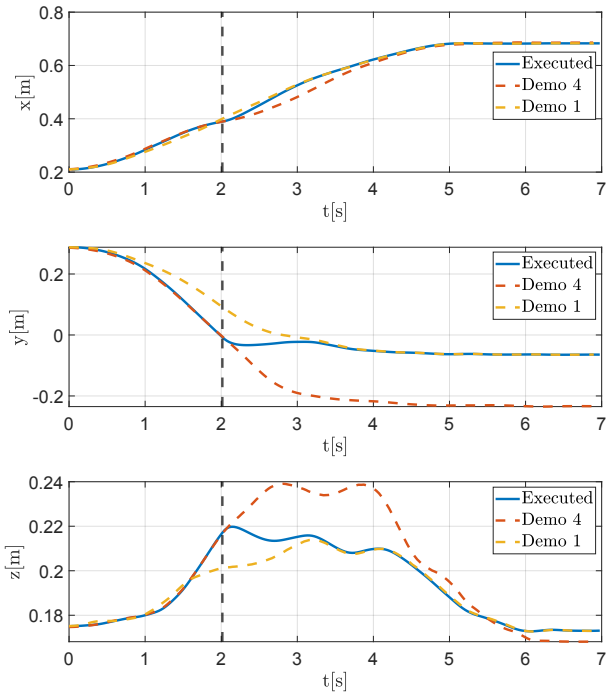

Fig. 9. Adjustment of the planned trajectory. The figure shows an example case during evaluation, where the HandNet recognized the human intention to be the same as the robot's ($k = 4$). The robot's goal slot was changed (to $k = 1$) and the DMP switched. All three components of the positional part of the robot's end effector trajectory are shown. Solid blue line shows the generated trajectory used on the robot, dashed lines denote demonstrated trajectories for the two relevant slots, and the vertical black line marks the time of the DMP switch.

into empty target slots, as seen in Fig. 1. To evaluate its applicability, the HandNet architecture was used as intention prediction method.

Robot Operating System (ROS) was employed to enable communication between the Intel RealSense camera, HandNet and the Panda robot. Four robot trajectories to each target slot were first recorded using kinesthetic guiding and encoded with DMP parameters (see Section III-D). The robot was sent to place an object into one of the slots by executing a recorded trajectory, while the human worker attempted the same. After the worker's motion started, the camera frames were being sent in real time to a computer, running an implementation of the HandNet architecture, where a forward pass with the input RGB-D video was carried out. The predicted probabilities of target slots $\mathbf{p}$ were then sent to the robot control system. Slot k with the maximum probability was selected as the intended target of the worker's motion and the robot reacted accordingly; if the predicted worker's target slot was the same or adjacent to the goal slot of the robot, the robot would change the goal slot to the farthest one possible (e.g., if the robot was moving towards slot $k = 2$, and the worker's intention was predicted as $k = 1$, the robot would switch to slot $k = 4$). The adjusting of robot motion in response to new predictions was implemented by switching from one DMP trajectory representation to another (as described in Section III-D).

An example of a collaborative task during the experiment is depicted in Fig. 8, where a sequence of images shows the process of human and robot motion and the adjustment of the planned robot trajectory due to HandNet intention predictions. The trajectory generated during this example can be seen in Fig. 9, denoted with blue color. We can observe a smooth and continuous transition between two demonstrated movements.

V. CONCLUSION

In this paper we proposed and evaluated a method for human intention recognition in collaborative tasks. We acquired a large database of human motions, consisting of marker-based position measurements and RGB-D videos, and used it to train two different networks to predict the final goal slot of the worker; OptiNet making predictions from position measurements and HandNet from RGB-D camera frames. Both networks achieved high accuracy in recognizing the human's intention. While the overall accuracy of OptiNet was higher, the implementation of HandNet in an industrial use case showed its viability. The relatively high price and complex setup associated with motion tracking systems, which are needed for OptiNet, makes the application of HandNet, which requires only an RGB-D camera, a convenient and effective approach.

The proposed RNNs alone are not sufficient to ensure

completely safe sharing of workspace between humans and robots. In industrial environments, an additional safety system must be installed to guarantee that there are no collisions or damage when the neural network predictions are not accurate enough. Most collaborative robots are already equipped with inherent limitations and sensors, which can provide the necessary safety, making them especially suitable for the presented approach.

An important enhancement to fully automate our approach would be a system to detect the beginning and the end of the human worker motion. For the recurrent neural network to compute accurate predictions, the sequential inputs must be similar to the training data, thus making it necessary for the network to start predicting when the human motion is initiated and stopping when it finishes. Further work could also include obtaining a larger, more diverse video-trajectory database of human motions with various objects for more accurate predictions. Another improvement would be the development of a high-level robot reasoning system to asses various situations and select the appropriate network for intention prediction.

Acknowledgment: This work has received funding from the program group Automation, robotics, and biocybernetics (P2-0076), and young researcher grant PR-09781, both supported by the Slovenian Research Agency, and from EU's Horizon 2020 grant CoLLaboratE (GA no. 820767).

REFERENCES

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

[2] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao, "Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly," *CIRP Annals*, 2020.

[3] Z. Wang, B. Wang, H. Liu, and Z. Kong, "Recurrent convolutional networks based intention recognition for human-robot collaboration tasks," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, 2017, pp. 1675–1680.

[4] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[5] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[6] A. Gams and A. Ude, "On-line coaching of robots through visual and physical interaction: Analysis of effectiveness of human-robot interaction strategies," in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 3028–3034.

[7] M. Simonič, T. Petrič, A. Ude, and B. Nemec, "Analysis of methods for incremental policy refinement by kinesthetic guidance," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, 2021.

[8] G. Hoffman, "Evaluating fluency in human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.

[9] A. Ude, "Robust estimation of human body kinematics from video," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyongju, Korea, 1999, pp. 1489–1494.

[10] Y. Li and S. S. Ge, "Human–Robot Collaboration Based on Motion Intention Estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.

[11] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao, "Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly," *CIRP Annals*, vol. 69, no. 1, pp. 9–12, 2020.

[12] M. S. Yasar and T. Iqbal, "A scalable approach to predict multi-agent motion for human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1686–1693, 2021.

[13] P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor, "Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018, pp. 5909–5914.

[14] M. K. Pan, V. Skjervøy, W. P. Chan, M. Inaba, and E. A. Croft, "Automated detection of handovers using kinematic features," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 721–738, 2017.

[15] T. Callens, T. van der Have, S. Van Rossom, J. De Schutter, and E. Aertbeliën, "A framework for recognition and prediction of human motions in human-robot collaboration using probabilistic motion models," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5151–5158, 2020.

[16] D. Widmann and Y. Karayiannidis, "Human motion prediction in human-robot handovers based on dynamic movement primitives," in *European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 2781–2787.

[17] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[18] R. Pahič, A. Ude, A. Gams, and J. Morimoto, "Deep encoder-decoder networks for mapping raw images to dynamic movement primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018, pp. 5863–5868.

[19] R. Pahič, B. Ridge, A. Gams, J. Morimoto, and A. Ude, "Training of deep neural networks for the generation of dynamic movement primitives," *Neural Networks*, vol. 127, pp. 121–131, 2020.

[20] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[21] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 8024–8035.

[23] G. Hinton, "Neural Networks for Machine Learning, Lecture 6e, rmsprop: Divide the gradient by a running average of its recent magnitude," https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012, [Online, accessed 22-October-2021].