Human action prediction in collaborative environments based on shared-weight LSTMs with feature dimensionality reduction

Tomislav Petković, Luka Petrović, Ivan Marković and Ivan Petrović

University of Zagreb Faculty of Electrical Engineering and Computing,
Department of Control and Computer Engineering,
Laboratory for Autonomous Systems and Mobile Robotics,
Unska 3, HR-10000, Zagreb, Croatia
{petkovic, luka.petrovic, ivan.markovic, ivan.petrovic}@fer.hr

Abstract

As robots are progressing towards being ubiquitous and an indispensable part of our everyday environments, such as home, offices, healthcare, education, and manufacturing shop floors, efficient and safe collaboration and cohabitation become imperative. Given that, such environments could benefit greatly from accurate human action prediction. In addition to being accurate, human action prediction should be computationally efficient, in order to ensure a timely reaction, and capable of dealing with changing environments, since unstructured interaction and collaboration with humans usually do not assume static conditions. In this paper, we propose a model for human action prediction based on motion cues and gaze using shared-weight Long Short-Term Memory networks (LSTMs) and feature dimensionality reduction. LSTMs have proven to be a powerful tool in processing time series data, especially when dealing with long-term dependencies; however, to maximize their performance, LSTM networks should be fed with informative and quality inputs. Given that, in this paper, we furthermore conducted an extensive input feature analysis based on (i) signal correlation and their strength to act as stand-alone predictors, and (ii) a multilayer perceptron inspired by the autoencoder architecture. We validated the proposed model on a publicly available MoGaze[1] dataset for human action prediction, as well as on a smaller dataset recorded in our laboratory. Our model outperformed alternatives, such as recurrent neural networks, a fully connected LSTM network, and the strongest stand-alone signals (baselines), and can run in real-time on a standard laptop CPU. Since eye gaze might not always be available in a real-world scenario, we have implemented and tested a multilayer perceptron for gaze estimation from more easily obtainable motion cues, such as head orientation and hand position. The estimated gaze signal can be utilized during inference of our LSTM-based model, thus making our action prediction pipeline suitable for real-time practical applications.

[1] https://humans-to-robots-motion.github.io/mogaze/

Keywords: human action prediction, long short-term memory networks, feature dimensionality reduction, correlation, autoencoder, gaze estimation

1. Introduction

With the robots becoming more capable and sophisticated, we are witnessing a growth in their presence and integration in private and professional human environments. Nowadays, such environments, besides cohabitation, often include close human-robot collaboration and interaction, yielding novel challenges concerning system efficiency and human safety. While robots are fully controllable, human behavior, on the other hand, although nearly optimal with respect to the task, is inherently stochastic. For example, imagine a healthcare worker treating a patient or a manufacturing shop floor worker assembling products in an agile production system. Their goals are well defined, but the execution and sometimes the environment are not completely controlled. While carrying out the task, the healthcare worker needs to adapt to the responses of the patient, while the worker on a manufacturing shop floor might change the order of the task execution for justified reasons. We argue that robots in human proximity should be aware of such changes and react accordingly. Having that in mind, one of the main challenges in collaborative environments is to capture the uncertainty and nuances of human behavior. Supervisory systems try to overcome these challenges by taking advantage of the plethora of methods that revolve around human trajectory prediction, safety regions assertion and action/goal prediction [1, 2, 3, 4, 5].

The problems of human action prediction and intention recognition have come under the spotlight of the research community in recent years. They serve as independent modules or are integrated into the human motion prediction either explicitly [6, 7] or implicitly [8]. The advantages of embedding human intentions implicitly in the model lie in the fact that those models can be trained jointly with the higher-level system and are validated straightforwardly through its performance. The higher-level system could be a fleet management system [9] that tries to reroute the robots out of a human's path and is evaluated by the warehouse deliveries, the number of rerouting, and collision number or a human trajectory prediction model [10] evaluated with the root mean square error of the predicted trajectory. On the other hand, explicitly estimating human actions enables the model to be crafted or trained independently of the higher-level system. In practice, this means that training the action prediction module can be done without the robots operating thus cutting costs. These models can also be interpreted more easily [11], allowing the higher-level system to have semantic meaning and reasoning of performed actions.

In recent years, human action prediction applications ranged from robotized warehouses [9, 12] to sedentary object-picking domain [13, 14, 15] and full-body motions [16, 17, 18]. State-of-the-art human action prediction frameworks are based on Markov models [19], inverse optimal control [11] or conditional random fields [20], which try to learn moving patterns with the respect to pertaining goals, usually assuming nearly-optimal human behavior in the observed sequences. In [5] the authors propose a hybrid deep neural network model for human action recognition using action bank features

leveraging fusion of homogeneous convolutional neural network (CNN) classifier. Input features are diversified and the authors propose varying the initialization of the weights of the neural network to ensure classifier diversity. Another approach based on the Long Short-Term Memory networks (LSTMs) is proposed in [21] where the authors craft a two-stream attention-based architecture for action recognition in videos. They suggest that such an approach resolves the visual attention ignoring problem by using a correlation network layer that can identify the information loss on each timestamp for the entire video. Furthermore, in [22] authors leverage a bidirectional LSTM to learn the long-term dependencies, and use the attention mechanism to boost the performance and extract the additional high-level selective action related patterns and cues. The convolutional LSTMS are used in [23] to handle the long-duration sequential features with different temporal context information and are compared to the fully connected LSTM. In [21] the authors propose an end-to-end two-stream attention-based LSTM network for human action recognition that selectively focuses on the effective features of the original input image. The concept of utilizing shared weights for neural networks was brought by de Ridder et al. in [24] with the focus on the feature extraction problem. This approach has gained traction in transfer learning [25] and physics simulation applications [26]. Regarding collaborative environments, the state-of-the-art models infer human actions by measuring different cues captured by wearable (eye gaze [27, 14, 28] or even heart rate and electroencephalography [29]) or non-wearable sensors. The use of non-wearable sensors such as motion capture systems or RGB cameras enables the model to capture crucial cues such as gestures [30], emotion [31], skeletal movement [32] or estimate eye gaze [33]. In works [28, 14, 34, 15, 35] authors have indicated that the eye gaze is a powerful predictor of human action. A good overview of human action prediction methods and their categorization by the type of problem formulation can be seen in [36]. Several works embed the eye gaze feature into human action prediction models using machine learning models such as support vector machine [14] or recurrent neural networks (RNNs) [34]. In the human collaborative scenario, the authors of [14] tested their algorithm relying on verbal instructions as additional features for their model and the actions form a sequence, In [15] the authors calculate the similarity between the hypothetical gaze points on the objects and the actual gaze points and use the nearest neighbor algorithm to classify the intended object. To the best of our knowledge, there does not exist a method that couples the human action prediction model with the directly measured eye gaze and human joint positions in a dynamic, changing environment. For example, in [14] the authors rely on gaze adding verbal commands in the feature space. In [15] the scenario is static and the subject sits while picking the objects who are always visible to the subject. Furthermore, in [36], the multiple-model estimator is leveraged for intention prediction, but the inputs to this model are extracted from a camera using convolutional networks and prior values that are not applicable in the dynamic collaborative domain.

In the last few years, multiple datasets concerning motion and action prediction have become publicly available but, to the best of our knowledge, none of them couple these two problems. Examples of purely motion prediction datasets are: ETH [37], KITTI [38] and UCY [39]. We encourage the reader to examine Table 2 in [40] for a detailed listing of the datasets and their descriptions. These datasets, alongside methods trained and evaluated on them [41], offer enough diverse data to train and test

human motion prediction models focused on answering the question *"Where is a human going to be during the next N steps?"*, but they are not adequately labeled with the context which would help to answer *"What is (the goal of) the observed human motion?*". On the other hand, datasets tailored for models focused on the second question, like the CMU's motion capture database [42], HumanEva [43] and G3D [44] excel in action diversity, but they are focused on distinguishing between different actions (jumping, catching, throwing), do not incorporate complicated motion patterns, and usually are not long enough for a long or mid-term human motion prediction problem. The MoGaze [34] dataset positions itself as an excellent blend of the aforementioned datasets because all the recorded motions have a labeled purpose (an object picking). Its subset has already been used by the authors for human motion prediction problems based on RNN networks and trajectory optimization [17, 45]. Therein, they used the Euclidean distance of the right hand to each object as an action prediction signal, improving their original motion prediction result. They also introduced the problem of graspability, which focuses on the exact wrist position at the moment of grasping, and placeability, defined as a probability distribution over possible place locations on a surface the carried object could be placed on. Mentioned models are not evaluated explicitly, but the authors compared a higher-level human motion prediction model's error for different graspability and placeability models thus validating them implicitly.

In this paper, we propose a novel human action prediction model based on shared-weight LSTM networks [46], a part of which was published in our preliminary work [47]. The novelty of the current paper with respect to [47] lies in the (i) expanded feature dimensionality reduction method, (ii) a new gaze estimation algorithm, (iii) exhaustive evaluation with an additional quality measure, and (iv) creation of a novel dataset that validated our approach as a general method for human action recognition. Similarly to related work, our model relies on the positions and orientations of human joints, recorded by a motion capture system, and on eye gaze captured using a wearable device, but with the following contributions: (i) to reduce the model complexity, we perform feature extraction through correlation and a multilayer perceptron inspired by the autoencoder architecture, (ii) architecture based on shared weight LSTM networks enabling dynamic adding and removing of human action goals, which is typical for collaborative environments, and (iii) since eye gaze might not always be available in a real-world scenario, we introduce a neural network-based gaze estimation that serves as an additional input to the proposed method and shows promising results. We have tested our approach on the publicly available MoGaze [34] dataset and published the code with a sample pretrained network. Additionally, we present SubMotion – a simpler dataset that includes six subjects, two female and four male, in object-reaching scenarios similar to the MoGaze. Our dataset records only the head orientation and hand position – a setup that could be easily applied in a real-world application without adding to workers' discomfort or costs. We compared the accuracy of the proposed model with alternatives such as the RNN network, fully connected LSTM network, and the strongest individual signal predictors (baselines), based on the area under the curve (AUC) score of the predicted goal accuracy and mean squared error (MSE) of the predicted goal location. Our model outperformed all of the baselines and alternative methods in MSE distance on both datasets and had better accuracy on the MoGaze dataset.

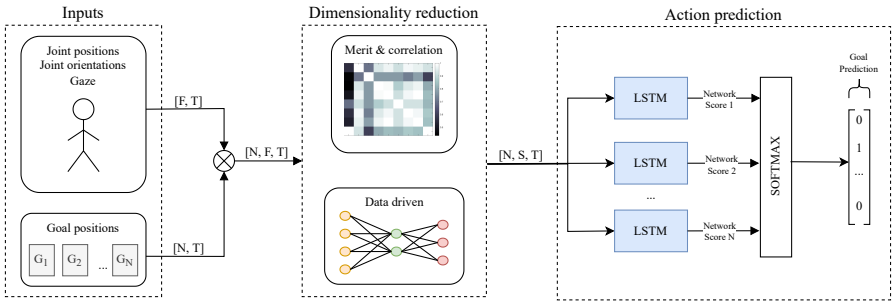


Figure 1: Pipeline of the proposed method. Square brackets denote the dimension of the corresponding tensor.

2. The Proposed Human Action Prediction Method

Our methodology follows the one published in our preliminary work [47] and is based on shared-weight LSTM networks and feature selection using correlation as well as feature extraction based on the autoencoder architecture. The goal of the proposed model is to ascertain which object in the environment will the human pick next. As we mentioned in the introduction, the creation of the MoGaze dataset with 1435 picking segments including the eye gaze, enabled us to craft a data-driven model for this problem. The segments are labeled with an ID of the object the human is going to pick and serve as ground truth for our framework. We design the proposed action prediction model as a general model for full-body motion that works in real-time and successfully captures relations between input cues and picked objects. Apart from that, we avoid learning specific relations between objects in a dataset. The main reason is that the objects can change their locations during operation and we want our model to handle a varying number of objects in a scene.

Another important aspect that needs to be taken into account by an action prediction model is long-term dependencies since goal inferring cues usually appear much earlier than the actual picking action [14]. For example, imagine a human that intends to pick a specific object from a shelf across the room. Prior to walking to it, they would probably look at that object to ascertain its location and path towards it. While walking, the gaze of the human would not be solely fixed on the object, but could also wander around the scene, especially if there are dynamic obstacles to be negotiated. Given that, a well-designed human action prediction model should take into account the fact that the gaze becomes fixed early in the sequence and can wonder thereafter. In other words, to successfully infer the goal, the model should be able to remember the most important past cue values, e.g., early gaze fixation at the object, as well as capture local tendencies, such as a human approaching the object. To achieve that, we propose multiple LSTM networks with shared weights to serve as the classifier for human action prediction.

However, relying on many inputs adds to the complexity and the network parameter number, which not only increases the run-time but can also impede the training process by increasing the risk of overfitting. Given that, we further introduce a feature selection method based on signal correlations and individual effectiveness to act as an

action prediction cue. To objectively validate our hand-picked selection of features, we also performed feature extraction with a multilayer perceptron (MLP) inspired by the autoencoder architecture and compared the manual selection with a fully data-driven approach.

The paper is organized as follows. In Section 2.1 we introduce the proposed shared-weight LSTM method for timely human action prediction. We also propose feature selection and extraction methods and the eye gaze estimation method. Section 3 brings the details of the SubMotion dataset we recorded for the purpose of testing the proposed framework. In Section 4 we do in-depth testing of the proposed framework on both MoGaze and SubMotion datasets and give detailed analyses of results. Finally, Section 5 concludes the paper.

2.1. Human action prediction framework

The proposed model fulfills two basic requirements: (i) to be fast enough so that the supervisory system can react in time and (ii) to have good generalization power. To address the latter, we crafted our model so that it can work in a changing environment and handle the addition or removal of objects in the scene. For example, in the MoGaze dataset, the objects are placed on three macro locations: two shelves and a table that do not move during the experiments. If we gave the model distances to all the goals as an input, the model could implicitly learn relations between those macro locations that would not hold should they move during the recording. Also, the number of objects in a scene could change and the transformation of a fully connected LSTM network to accommodate this circumstance would not be a trivial task.

Having that in mind, we decided to approach this problem by training a single classification model and our framework is illustrated in Fig. 1. For each observed sequence of length T we gather the following input features F: joint positions that are used to calculate Euclidean distances towards each of the N goal positions in the dataset, and gaze and orientation unit vectors that are used to calculate the Euclidean distance between them and the unit vector pointing towards the position of an object. All features are normalized based on the average value in the training set. Each of N sequences is labeled with 1 if it belongs to the object that is eventually going to be picked, otherwise, it is labeled with 0. All the sequences in the training set are aggregated and the dataset is balanced by randomly removing sequences that belong to the "not-a-goal" class. Finally, we train a single LSTM network model for sequence classification with a softmax activation on this data. Note that, during the training, the model does not have access to absolute orientations and positions of the joints or the goals. As a consequence, it learns only if the observed feature sequence (relative to an object) belongs to a pertaining goal or not.

During runtime, we evaluate all selected features for each of the N goals and send them as inputs to N LSTM networks with shared weights (feature selection and extraction are explained in Section 2.3). For the MoGaze dataset, the number of goals was $N = 10$, while for our novel dataset it was $N = 5$. We aggregate outputs of each network via softmax [48] activation function and select the goal whose network has the highest score. This approach enables us to easily add or remove goals if they change during operation which was an important reason behind training only a single LSTM model. Furthermore, by training only a single model that receives relative distances

as input and classifies whether that input sequence of features is the pertaining goal or not we remove any contextual environment location information. For example, in the MoGaze dataset, the objects are placed on three macro locations, two shelves, and a table, which do not move during the experiments. If we give the model, e.g. distances to all the goals as an input, the model could implicitly learn relations between those macro locations that would not hold for other datasets. By utilizing the shared weight concept, we ensure the decision-making process for each goal is the same.

2.2. Preliminaries on LSTM networks

An RNN, introduced by Rumelhart et al. [49], is a neural network that consists of a hidden state $\boldsymbol{h}_t$ which is connected to its output $\boldsymbol{y}_t$ as well as previous hidden state $\boldsymbol{h}_{t-1}$. This property enables it to capture a temporal dynamic behavior of the process and propagate the information through time because its output depends both on the input at a given time step as well as on the hidden state at the previous time step. Formally, a basic RNN can be described by:

$$\boldsymbol{h_t} = \sigma(\boldsymbol{U_h}\boldsymbol{h}_{t-1} + \boldsymbol{W_h}\boldsymbol{x}_t + \boldsymbol{b_h}) \tag{1}$$

$$\boldsymbol{y}_t = \sigma(\boldsymbol{W_y}\boldsymbol{h}_t + \boldsymbol{b_y}) \tag{2}$$

where σ is the sigmoid, the most commonly used activation function, and $\boldsymbol{x}_t$ is the network input. The next hidden state is calculated using hidden weights $\boldsymbol{U_h}$ and input weights $\boldsymbol{W_h}$ while output weights $\boldsymbol{W_y}$ are used for calculating the output. The model also incorporates hidden layer and output bias - $\boldsymbol{b_h}$ and $\boldsymbol{b_y}$. RNN networks have been successfully used in a plethora of time-series prediction problems, including action sequence prediction [16]. However, the main deficiency of the RNN model is the vanishing and exploding gradient problems for longer sequences making it unfit for problems that have long-term dependencies that need to be captured.

The LSTM networks, introduced by Hochreiter et al. [46], is a derivative of RNN networks with the introduced cell state and gates, formally:

$$\boldsymbol{f_t} = \sigma(\boldsymbol{U_f}\boldsymbol{h}_{t-1} + \boldsymbol{W_f}\boldsymbol{x}_t + \boldsymbol{b_f}) \tag{3}$$

$$\boldsymbol{i_t} = \sigma(\boldsymbol{U_i}\boldsymbol{h}_{t-1} + \boldsymbol{W_i}\boldsymbol{x}_t + \boldsymbol{b_i}) \tag{4}$$

$$\boldsymbol{o_t} = \sigma(\boldsymbol{U_o}\boldsymbol{h}_{t-1} + \boldsymbol{W_o}\boldsymbol{x}_t + \boldsymbol{b_o}) \tag{5}$$

$$\boldsymbol{g_t} = \tanh(\boldsymbol{U_g}\boldsymbol{h}_{t-1} + \boldsymbol{W_g}\boldsymbol{x}_t + \boldsymbol{b_g}) \tag{6}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \boldsymbol{g}_t \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \sigma(\boldsymbol{c}_t) \tag{8}$$

$$\boldsymbol{y}_t = \sigma(\boldsymbol{W_y}\boldsymbol{h}_t + \boldsymbol{b_y}) \tag{9}$$

where tanh is the hyperbolic tangent function and the operator $\circ$ is the Hadamard product. The cornerstone of an LSTM model is the *cell state* $\boldsymbol{c}_t$ that propagates information through time. Every iteration of the LSTM network first forgets irrelevant information in $\boldsymbol{c}_t$ using the *forget gate* $\boldsymbol{f_t}$, and then adds new information with the *input gate* $\boldsymbol{i_t}$. The $\boldsymbol{c}_t$ is then used for future iterations of the LSTM network as well as for updating current *hidden state* $\boldsymbol{h}_t$ using the *output gate* $\boldsymbol{o_t}$. Finally, the output of any given iteration is

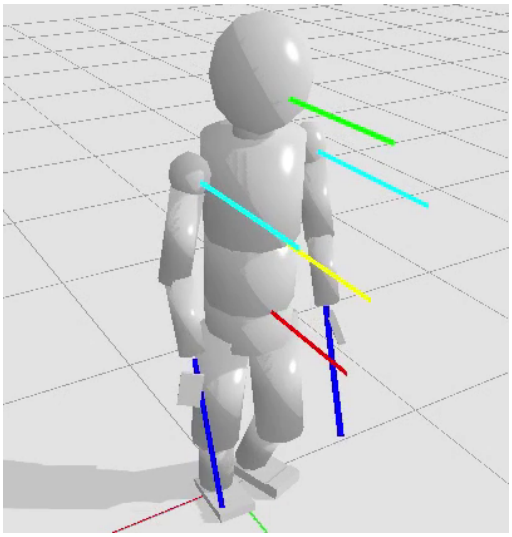


Figure 2: Orientations assigned to joints. Head, torso, pelvis and shoulders orientation is selected to match forward looking direction while hand orientations are selected to match forearm direction.

calculated in the same manner as in the RNN model. LSTM networks have seen applications in similar problems, such as pedestrian trajectory prediction [50], and we have selected them as our backbone due to their ability to capture long-term dependencies.

2.3. Feature dimensionality reduction

In previous sections, we discussed the proposed model based on LSTM networks for human action inference. For our application, the input features for our model are time series of human joint positions and orientations as well as the eye gaze of the subject. The proposed framework processes these features by numerous matrix additions and multiplications. Each additional input feature adds to the dimensionality of these matrices, thus increasing the number of operations and execution time. Moreover, it could potentially also create the need for increasing the number of hidden dimensions in the network architecture. This is certainly an unwanted side effect, not only for previously stated reasons but also due to the limited amount of training data. Having this in mind, we assert that it is important to craft a feature dimensionality reduction method that will indicate which of the recorded joint orientations and positions should be the most relevant inputs to our model. To solve this problem, we took two different approaches.

The first is based on time series analysis - it uses signal correlations to ascertain similarities between features. Our intuition is that features that correlate highly can be substituted by only a single feature from that group. This approach was first proposed by Hall et al. in [51] as used in [51, 52, 53]. In order to choose the most representative feature of the group, we have ranked each feature using the area under curve (AUC) score and selected the highest-ranking feature. The AUC for each feature is calculated for a time span of three seconds (360 frames), as proposed in [34], using an average of accuracy curves on the train set. The accuracy curves are obtained for each feature

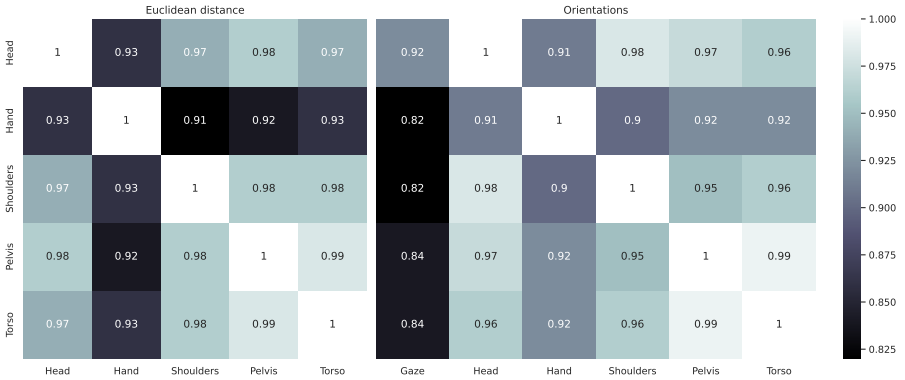


Figure 3: Correlations of selected input features. The Euclidean distances are in the left part of the table, while gaze and orientations are in the right part.

by checking if the joint is closer to the actual goal than to any other object (Euclidean distance) or if the difference between the orientation vector of a joint and a vector from which that joint sees the object smallest for the goal (orientation distance).

At this point, it is important to clarify the method we used for extracting the joint orientations because the authors give them relative to the humanoid configuration's initial pose, while we need to use them with respect to the world scene. We decided to use a "T-pose" with a human looking towards the x axis as the initial configuration and define all orientations of joints in that pose as $[1, 0, 0]$. This way we ensure that orientations of the head, torso, and pelvis tend to align with the motion direction which we argue is an intuitive way to define orientations of the joints given our application. An example of orientations assigned to joints of interest is shown in Fig. 2.

Now that we have all set up for feature selection, we calculated correlations between all of the feature distances towards all the goals, and the comparison can be seen in Fig 3. Euclidean distances of all the joints correlate highly and less so with the hand because of the reaching motion. Orientations of all joints also correlate highly, but less so with the hand orientation. Furthermore, the gaze correlates weakly with all the other features, except the head indicating that the head orientation could be useful in a gaze estimation problem (when no dedicated gaze tracking equipment is available). The correlation analysis implies that a good subset of features would include the eye gaze, hand position, and orientation of one of the following joints: head, shoulders, pelvis, and torso. We have then proceeded to calculate the AUC score of the proposed input features which we henceforth call baselines since we see each as a potential sole feature for action prediction. Theoretically, a good data-driven model should score better than any single feature, i.e., than any baseline. Finally, the eye gaze scored 155.0, head orientation 71.4, and hand position 83.7, and they were selected as input features for our model. Other baselines scored significantly less than 70. One can notice that we did not analyze joints like toes, knees, and elbows. The main reason is that they correlated poorly with each other and scored very low on the AUC metric which supports our intuition that these joints are of less importance for our application.

The second approach we took is based on autoencoders. Autoencoders are multilayer perceptrons (MLPs) with two main parts: an encoder that maps the inputs to the hidden layer or codes the inputs, and a decoder that reconstructs the input from the hidden layer. If the hidden layer is large enough, the autoencoder can completely recover the input signal at the output. However, in practice, the dimension of the hidden layer is usually much smaller thus forcing the autoencoder to approximate the input by preserving only the most significant information contained within. Because of that, autoencoders are widely used in feature extraction [54, 55] and selection [56, 57] applications.

We have followed this intuition behind autoencoders and implemented an MLP-based feature extraction. The proposed MLP has an architecture similar to an autoencoder with an input layer that takes all the recorded joint positions and orientations, which is then followed by one hidden layer of a smaller dimension. Finally, the output layer consists of three fully connected neurons, since we wanted to match the number of features used by the correlation-based feature selection method. We tested all commonly used activation functions such as hyperbolic tangent and ReLu [58], and decided to use the sigmoid function as it demonstrated the best performance. Unlike the vanilla autoencoder, our data-driven feature extraction MLP is not trained to match the input data, but is directly connected to the backbone network and trained in an end-to-end fashion. The intuition behind this approach was to enable the training process to refine useful information using data. The extracted input features are composed of a linear combination of all feature input candidates and don't perfectly match any of them. However, by comparing results with the hand-picked features we are able to validate our merit-based approach. The parameter number of the entire model is also reduced because the addition of the fully connected MLP is outweighed by reducing the input dimension of the backbone network.

2.4. Gaze estimation

Even though the eye gaze has proven to be the most accurate baseline for human action prediction, it might not always be available in real-time practical applications. It requires the user to wear it on their head the whole time, which can be inconvenient and hinder the person's task execution, especially when performing complex tasks. However, the absence of gaze measurements would make our inference with the proposed shared-weight LSTM networks unviable, since we trained the model to expect gaze in the input along with other motion cues. To alleviate this issue, we propose to estimate the eye gaze from other, more easily obtainable features, such as head orientation and hand position.

Head orientation and hand position can be obtained in real-time from practical wearable sensors, e.g., IMUs mounted on a person's helmet and watch [59, 60]. While the problem of gaze estimation might seem intractable in the general case, due to the human eye gaze presenting an additional degree of freedom compared to the head orientation, we assert that the hand position in collaboration tasks might provide additional information that correlates significantly to the eye gaze. For example, if a person is reaching for an object with their hand, our assumption is that the person will also be looking towards the object in question, thus connecting the gaze to the other motion cues. The proposed estimation procedure relies on having a dataset of human motion

while wearing the gaze tracking equipment and then employing a data-driven model to capture the mapping of the subject's head orientation and hand position to their eye gaze. Our gaze estimation model is an MLP that consists of three layers, where the hidden layer is of dimension 10 and the activation function is a rectified linear unit (ReLU). The inputs of our network are hand position and head orientation vectors, while the output of the network is the eye gaze vector. We trained our network using stochastic gradient descent. Once the model is learned, it is utilized during test time to infer the gaze which is then used as an input to the LSTM networks. In practice, this would mean that we can perform a one-time recording of the worker's gaze during collaboration tasks, learn the gaze estimation model using that data, and then perform action prediction during future runs in real-time without requiring the worker to wear the uncomfortable gaze tracking equipment. Potentially, an "average model" could be learned across multiple participants that could generalize well to other people for the same tasks, but this question is out of the scope of current research.

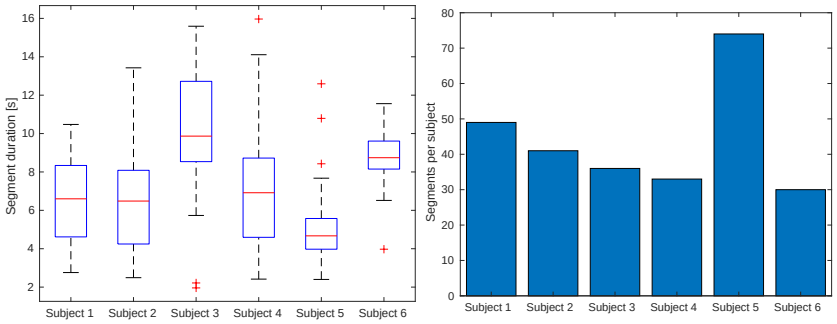
3. The novel SubMotion dataset

In order to demonstrate the general application of the proposed algorithm, we have recorded our own dataset which aims to complement the much more comprehensive MoGaze dataset. Unlike the MoGaze dataset, which uses a specialized recording suit and proprietary software to obtain the configuration of the entire human body, our dataset records the positions, and orientations of only two joints: the head and (right) hand. Since it includes only a small subset of human motion features, we dubbed it the SubMotion dataset. Furthermore, the SubMotion setup could be easily embedded in a real-world application without adding to the worker's discomfort. While we have also recorded our data using the OptiTrack system, position of hand and orientation of head could potentially be extracted with wearable sensors in workers' helmets, gloves, or watches. We recorded six times less amount of data than the MoGaze dataset to demonstrate that the proposed algorithm can be trained without the abundant amount of data. This section describes the dataset recording setup and the method we used to obtain the segments we trained and tested our model on.

3.1. Experimental setup

We have used the OptiTrack motion capture system with 12 Flex13 cameras covering the entire workspace. Human participants wore a helmet and a glove with reflecting markers that captured the head and hand locations and orientations. We have chosen a minimal set of wearable equipment which can be easily worn by a worker in a collaborative human-robot scenario without impeding their efficiency or causing discomfort and fatigue. The workspace consisted of three tables on which five objects were placed with an obstacle in the middle. Unlike in the MoGaze dataset, objects in our dataset are static and we don't need to track their position during the recording.

We have recorded the experiment with a total of six subjects, two female and four male, in an object-reaching scenario. Subjects also varied in height, ranging from 155cm to 195cm. Each subject was introduced to the elements of the scene and shown the position of each object. This step was particularly important because the exact



(a) Segment durations per subject.

(b) Total amount of segments per subject.

Figure 4: The SubMotion dataset analysis. We compensated lower average segment duration of Subject 5 by recording more segments to balance the dataset.

positioning of the helmet on the subject's head can vary between subjects. The MoGaze dataset takes advantage of OptiTrack's software for full-body tracking which yields orientation of the head as a property of the obtained human body configuration. In our case, the helmet is defined as a rigid body and its orientation is relative to the orientation the helmet had at the initialization time. Since the head orientation is a crucial input feature for the proposed method, we needed to calibrate its orientation for each subject. We instructed the subject to look at each object at the beginning of the experiment and thus were able to extract reference orientations of the helmet corresponding to each object. This data was used to calculate the helmet's transformation matrix for each subject using MATLAB's ABSOR [61] tool for least-squares estimation of the rotation based on the Horn's [62] quaternion-based algorithm.

3.2. Dataset Recording

After the described initialization phase, we recorded two scenarios per subject. In the first scenario, each subject began the recording segment at the same starting point where they waited for the instruction on which object to pick. After the instruction, the subject identified the object, moved to its proximity, picked it, and placed it back on its spot. Subjects were instructed to pick the objects using only the hand that has been recorded. Then they returned to the starting point and waited for the next instruction. We have generated the order of instructions randomly ensuring that each object is picked an equal number of times. In the second scenario, subjects were allowed to walk freely in the scene. Once they decided which object they are going to pick next, they communicated their intention and carried on to execute it as in the first scenario.

We have recorded a total of 30 minutes of data at 120 FPS which is 6 six times fewer than the amount of data present in the MoGaze dataset. The data was split into segments for each subject and the segments were labeled with the object that is eventually going to be picked. The starting point of each segment is when the subject would reach the starting position in the first scenario or when they would communicate the intention in the second scenario. The final point is the moment when the object gets picked. We have analyzed the distribution of segment lengths and the total amount of

segments per subject as can be seen in Fig. 4. We can see that, on average, we recorded more than the three seconds per segment for each subject, which is important because the proposed algorithm is evaluated on the last three seconds of each segment. The SubMotion dataset can be made available on request.

4. Experimental Results

In this section we present and discuss results of the proposed method on two datasets: the MoGaze and SubMotion. Unlike our previous work in [47], where we trained a model on data belonging to one half of the subjects and performed testing on the other half, in this paper we decided to train a unique model for each subject. Such a decision was motivated by observing different motion patterns and data capture quality between subjects, which manifested mostly on the eye gaze. Also, as body proportions, gait and behavior patterns tend to be rather individual, it seems natural to assume that the action prediction models will work better if they are individually trained. our evaluation was performed by k-fold cross-validation in order to demonstrate the statistical significance of our results. In practice, we randomly partitioned the data for each subject in the MoGaze dataset into five equally sized subsamples, while the SubMotion dataset was partitioned into three subsamples. Each model was tested on one subsample and training was done on the union of the rest.

We tested multiple proposed configurations including the multiple RNN and LSTM networks with shared weights as backbones, with and without MLP for feature extraction. We have explored the option of using one LSTM with information about the entire scene as an input and we tested for different dimensions of the hidden layer to obtain the best possible result. The configurations were compared in three quality measures:
i) Area under Curve: Following our previous work, we continued to use the AUC score as a scalar value representing the accuracy of a model. It is calculated as the average accuracy for each time step in the three-second evaluation window.
ii) Mean Squared Error (MSE) While the AUC score shows in how many frames the proposed method guessed the right goal, it fails to encapsulate how much of the method was when it got the goal wrong. For example, guessing the wrong goal which is 15 cm from the right goal is not the same as guessing the goal which is 1 m away. Having that in mind, we introduce the normalized MSE of the expected goal location for each frame as:

$$\text{MSE} = N\frac{\|l_g - \sum_{i=1}^{i=N} p_i l_i\|}{\sum_{i=1}^{i=N} \|l_g - l_i\|} \tag{10}$$

where l_i is the location of i-th object, l_g is the location of the goal object, N is the number of objects and p_i is the probability that the i-th object is the goal and is calculated as the output of the corresponding network divided by the sum of all the network outputs. We use the average distance between objects as the normalization factor.
iii) Execution time: We tracked the average execution time for each of the proposed models to ascertain if they are sufficiently computationally efficient and could enable a potential supervisory system to react accordingly.

Our size of the evaluation window follows from [34] and equals three seconds (360 frames). All of the models were trained and tested on the Intel Core i7-7700HQ CPU.

The main reason why we decided to use a CPU rather than a graphical processing unit was to show that the proposed model is indeed lightweight and can be easily incorporated into any supervisory system without additional hardware dependencies. The models were implemented in Pytorch and we have made the backbone network publicly available[2]. We trained the model for 100 epochs and used the batch size of 64 for MoGaze and 16 for SubMotion, leveraging Adam [63] optimizer with the learning rate of 0.01. The forementioned parameters were obtained experimentally via exhaustive testing.

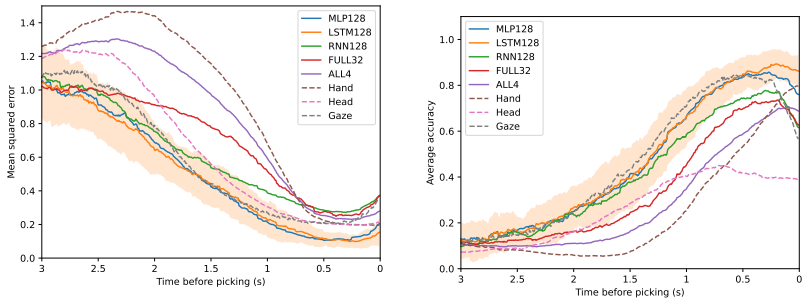
4.1. MoGaze results

We compared the results of the selected baselines, namely the gaze, head orientation, and hand distance, with the following neural network models:

- LSTM*n*: denotes multiple LSTM classifiers with shared weights and hidden dimension *n*. The input for these classifiers are features selected by their individual effectiveness and correlation as in [47]. The best result was achieved for LSTM128.
- MLP*n*: denotes multiple LSTM classifiers with shared weights and hidden dimension *n*. The input for these classifiers is all features that first pass through an MLP feature extraction with the hidden dimension of $1 - 8$. The best result was achieved for MLP128 with a hidden dimension of 2.
- RNN*n*: denotes multiple RNN classifiers with shared weights and hidden dimension *n*. The input for these classifiers are features selected by their individual effectiveness and correlation as in [47]. The best result was achieved for RNN128.
- FULL*n*: denotes an LSTM that takes the selected features for all objects as input and uses a softmax layer to perform classification of the estimated goal. The best result was achieved for FULL32.
- ALL*n*: denotes an LSTM that takes all available features as input and uses a softmax layer to perform classification of the estimated goal. The best result that was in accordance with the run time constraints was achieved for ALL4.

We evaluated baselines by interpreting the inverse of the distance towards each goal as the score at any given time point and treating these scores in the same manner as the network outputs to obtain the prediction. The results of the 5-fold cross-validation can be seen in the Fig. 5 and Table 1. The eye gaze has proven to be the best performing baseline with almost double the AUC score compared to the head orientation and hand distance baselines. This result is in accordance with our previous findings which indicated that the gaze baseline acts as the strongest predictor for the object that the human is going to pick, and furthermore, it can distinguish the actual goal among the nearby objects with pinpoint accuracy [47]. The eye gaze also had the smallest MSE but with a smaller margin indicating that subject often fixated their gaze at the goal but was also often browsing around the environment.

[2] `https://github.com/petkovich/ensemble-lstm`



(a) Mean squared error of expected goal location before picking the object.

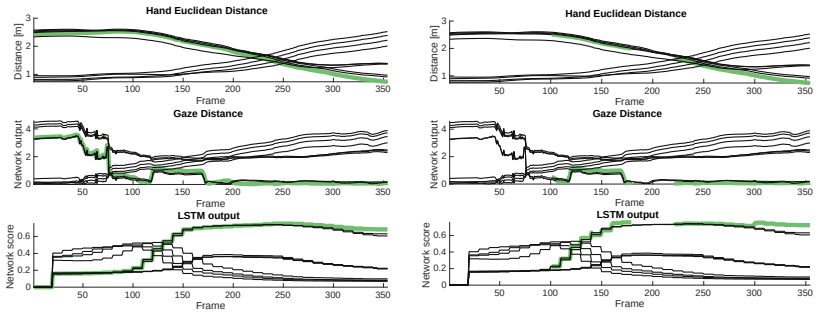
(b) Average accuracy before picking the object.

Figure 5: Average values of the MoGaze cross-validation. Additionally, standard deviation of the best performing model (LSTM128) is highlighted.

Our proposed model based on the shared-weight LSTM networks outperformed in AUC all the baselines and succeeded to beat the gaze by 1.1%. Following the argument presented in [47], we claim that it is hard, if not impossible, to beat the eye gaze significantly in this quality measure. On the other hand, our model had smaller MSE than the gaze baseline by 14.7%. This means that our method, on average, missed the actual goal location by 0.49 times the average distance between all objects while the eye gaze missed it by 0.57. If we imagine a supervisory system that has to reroute a robot to help the human with executing a task at the goal location, a smaller estimated goal location error could lead to better efficiency of the system. The shared-weight LSTM networks outperformed the shared-weight RNN networks demonstrating LSTM superiority in this time series classification problem. It also achieved a much better result in the full LSTM network, which was expected having in mind that the positions of the objects change during the recording. The use of MLP did not have a positive effect on the result in this case, implying that our feature selection method helped not only to reduce the complexity and execution time of the model, but also to improve the result. Execution times suggest that the proposed framework works at 400 Hz on the CPU which implies it can be seamlessly integrated into the decision-making loop with modern-day sensors. For example, MoCap systems used for recording of both datasets

	AUC	MSE	Execution Time [ms]
Gaze	168.0	206.3	-
Hand	87.1	339.0	-
Head	96.0	237.8	-
LSTM128	**169.9**	**175.9**	1.7
MLP128	166.6	178.2	2.5
RNN128	149.5	218.2	**0.9**
FULL32	124.4	248.5	2.1
ALL4	102.9	304.6	4.1

Table 1: MoGaze dataset results.



(a) (b)

Figure 6: The goal adding and removing experiment. The left figure shows the output of the network with 10 goals that are not changed during the experiment. The right image starts with 9 goals and the eventually picked object is added at frame 100, removed at frame 180 and finally added at frame 220. The network quickly adapted to the addition and removal of the goal and finally succeeded to infer the picked object.

were running at 120 Hz.

One of the main advantages of the proposed framework is its ability to quickly adapt to dynamic and unknown environments. For example, in the collaborative environment, a new interesting item can appear during the operation. Also, some items that have previously been present in the scene can disappear, i.e. they can break, be consumed or become unnecessary. Because of that we have implemented and tested the capability of the proposed framework to handle adding and removing objects (goals) from the scene. Removing the potential goal is done trivially by removing the input connection to the LSTM in Fig. 1 forcing its output to 0. Adding a goal is done in a similar manner, by attaching a new copy of the same LSTM to the action prediction pipeline. The hidden state $\boldsymbol{h}_t$ and cell state $\boldsymbol{c}_t$ of the attached LSTM are inherited from the object closest to the new object at the time of adding. We have also tested initializing the LSTM states with zeros and random values but the proposed method has shown the best results. We have tested adding and removing goals on several experiment runs and the example of one is shown in Fig. 6. It is important to note that fully connected action prediction models such as the FULL32 do not have the capability to reduce or expand input dimension.

4.2. SubMotion results

We continued with experimental validation of the proposed framework on our SubMotion dataset described in Section 3. In this case, we compared the network results to the baselines: head orientation and hand distance with models defined as in the previous section using the same abbreviation conventions. Since we have fewer data per subject at our disposal, we decided to reduce the degree of the cross-validation to three. For the same reasons, model complexity has been scaled back and generally, the best performing models had two to four times smaller hidden dimensions than the best corresponding MoGaze model. Also, for this application, the MLP feature extraction had a hidden dimension of 1. The results of the 3-fold cross-validation can be seen in Fig. 7 and Table 2.

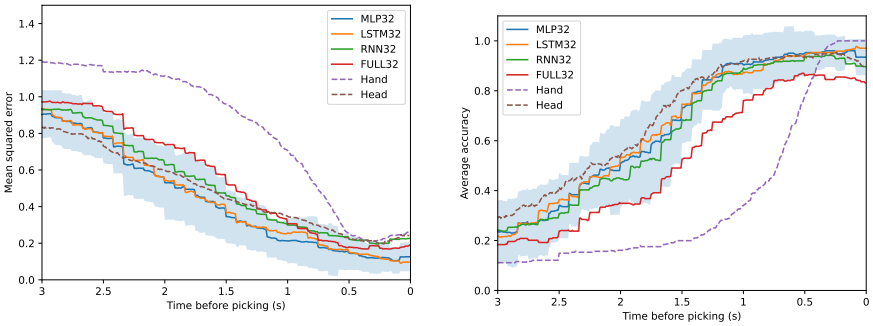
One can notice that the best performing baseline on this dataset is the orientation of the subject's head, similarly to the performance of the gaze on the MoGaze dataset. However, it has proven to be a much stronger cue for action prediction than the same feature in the MoGaze dataset. The exact cause of such a phenomenon is unclear but there are a few possible explanations we would like to mention. Firstly, we calibrate the position of the helmet on each subject individually as described in the Section 3, while the joint orientations on the MoGaze dataset follow from the predefined human body configuration and are prone to change depending on the exact fit of the marker suit on each subject. Secondly, it is possible that subjects are aware that eye gaze and position of particular joints are measured which can lead to the manifestation of the Hawthorne effect [64] introducing a bias into both datasets. Having that in mind, we proceed carefully with the interpretation of obtained results.

As the only additional feature for our model, apart from the head orientation, is the Euclidean distance of the right hand from all objects, no method was able to beat the head orientation baseline on the SubMotion dataset. Similar to the MoGaze dataset results, the LSTM model outperformed the RNN model and the shared-weight LSTM networks have performed better than the full LSTM network. On the other hand, the MSE analysis has once again shown the advantages of the proposed model which outperformed the head baseline by 8.8%. The MLP embedding produced an even better result, outperforming the baseline by 11.3%. This is a promising result showing that, although the proposed model is not correct most of the time (lower AUC), it produces smaller errors distance-wise (lower MSE). The full LSTM network performed poorly even though the objects did not move between the segments in this dataset, which additionally justifies the use of the shared-weight method. Compared to the more complicated models used on the MoGaze dataset, execution times were reduced and small enough to ensure real-time operation.

The main motivation for recording the SubMotion dataset was to complement the MoGaze dataset and show that similar results can be achieved using a much smaller and lightweight setup. Furthermore, we wanted to explore the effect of transfer learning approaches by training a single human action prediction model on the MoGaze dataset and testing it on the SubMotion dataset. Unfortunately, we were unable to achieve any sensible result with such an approach which probably follows from the previous argument about differences between the head orientations on these datasets. We tried to leverage gaze in the MoGaze dataset as a comparable signal to the head orientation in the SubMotion dataset but the dynamics of these signals differ a great deal which

	AUC	MSE	Execution Time [ms]
Hand	129.9	296.2	-
Head	**253.1**	173.5	-
LSTM32	240.9	158.2	1.1
MLP32	240.2	**153.9**	1.2
RNN32	229.0	191.8	**0.7**
FULL16	191.8	197.7	1.9

Table 2: SubMotion dataset results.



(a) Mean squared error of the expected goal location before picking the object.

(b) Average accuracy before picking the object.

Figure 7: Average values of the SubMotion cross-validation. Additionally, standard deviation of the best performing model (MLP32) is highlighted.

rendered such an approach invalid.

4.3. Gaze Estimation results

In this section we report the performance of the proposed shared-weight LSTM networks when coupled with the eye gaze estimation procedure proposed in Section 2.4. Evaluating our action prediction LSTM networks model with the estimated gaze, which we dubbed EG-LSTM, required us to partition the MoGaze dataset into three sets. The first set was used for gaze estimation training, meaning that we fed the head orientation and hand position signals as inputs to an MLP proposed in Section 2.4 and used the recorded gaze as a supervisory signal during learning. Then we utilized the learned model to estimate the gaze signal on the second and third set from the head orientation and hand position. We calculated the MSE between ground-truth gaze data and gaze predictions, with the average MSE on the second and third set evaluating to 0.0621. The second set with the estimated gaze was then used as a training set for the shared-weight LSTM networks, while the third set was used for evaluating the shared-weight LSTM networks with the estimated gaze. This way of partitioning the datasets is transferable to real-world applications. If there exists a pre-recorded dataset that contains gaze measurements, it can be used to train the proposed MLP. Then we can infer the gaze estimates during a person's activity in real-time when the gaze measurement is unavailable and use that data to train and infer with the shared-weight LSTM networks. We compared the average accuracy of EG-LSTM before picking the object with the shared-weight LSTM networks trained with head orientation and hand position as well as the hand, head, and estimated gaze baselines. The results of our analysis are depicted in Fig. 8. In our evaluation, the EG-LSTM achieved the AUC score of 138.26, outperforming the LSTM trained with head orientation and hand position by 16%. This implies that the estimated gaze signal contained additional information that was utilized in learning the EG-LSTM model to achieve better average accuracy. Its performance matched the ground-truth gaze baseline, having the AUC score within 0.5%, although the accuracy curves were qualitatively different. Gaze baseline is more accurate at an

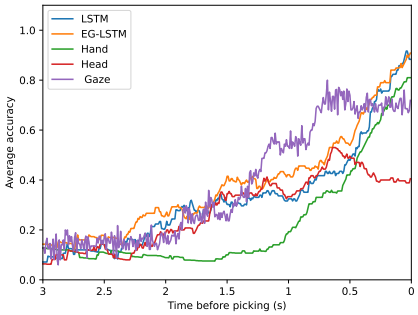


Figure 8: Average accuracy before picking the object. The EG-LSTM outperforms the LSTM model without estimated gaze as an input feature.

earlier stage of about 1 s before picking the object, while EG-LSTM was more accurate in the last half of a second before picking. This behavior is consistent with results from earlier sections, where the LSTM relied on the hand position motion cue in close proximity to the goal. Our findings demonstrate that the shared-weight LSTM networks have the potential to work well in specific situations even when human eye gaze measurements are unavailable, which is practical for many real-world applications.

5. Conclusion

In this paper we have introduced a human action prediction framework based on shared-weight LSTM networks and feature dimensionality reduction. The idea behind our framework was to enable a supervisory system or a robot to have a timely and efficient reaction to accurately inferred human actions. For this paper, we decided to focus on the object picking problem, where we strived to predict which object in the scene the human is going to pick next since this represents a strong proxy of typical human-robot collaboration tasks and can be elegantly validated.

We have carefully analyzed MoGaze, a publicly available dataset that captures long sequences of full-body everyday manipulation tasks along with the subject's eye gaze. As the MoGaze dataset captures the motion of 21 joints, we crafted a feature dimensionality reduction method based on correlation and presented an alternative method based on a multilayer perceptron inspired by the autoencoder architecture. The MoGaze dataset was recorded using a specialized body suit with reflective markers and eye gaze tracking hardware. In order to demonstrate the general applicability of our method, we created a smaller dataset recording only the orientation of the subject's head and the position of their hand – a setup that can be more easily incorporated into workers' helmets, gloves, or wearable watches.

The proposed model consists of multiple LSTM networks with shared weights trained to classify whether or not the observed sequence relative to an object is the goal or not. By comparing the output of all LSTM networks during runtime we infer which object the subject is going to pick next. This model follows from our previous work [47] with a novel autoencoder embedding as an additional feature processing unit.

Furthermore, we have implemented and tested a multi-layer perceptron for estimating gaze from more easily obtainable motion cues such as head orientation and hand position. The estimated gaze signal is then utilized during inference of our LSTM-based model in cases when gaze measurements are unavailable, which can often happen in practical applications.

We have validated the proposed models exhaustively on both datasets and compared them with the baselines, RNN networks, and the LSTM network that has locations of all the goals as the input. We measured the area under curve score, which represents the total accuracy of the model, and the mean squared error, which shows how much on average we missed the expected location of the goal and the execution time of our model. The results have shown that the eye gaze is the most powerful cue for human action prediction problems followed by the orientation of the head if the eye gaze is not available. Our method succeeded to beat the baselines in the MSE on both datasets and AUC on the MoGaze dataset. It is computationally efficient enabling it to run in real-time on a standard laptop CPU. We also tested the proposed model coupled with the eye gaze estimation procedure, demonstrating that the gaze estimate improves the model performance, when compared to using only the head and hand motion as features, thus enabling better performance when gaze measurements are unavailable.

Acknowledgment

This research has been supported by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS).

References

[1] M. V. da Silva, A. N. Marana, Human action recognition in videos based on spatiotemporal features and bag-of-poses, Applied Soft Computing 95 (2020) 106513.

[2] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, J.-K. Kämäräinen, Ar-based interaction for human-robot collaborative manufacturing, Robotics and Computer-Integrated Manufacturing 63 (2020) 101891.

[3] T. B. Pulikottil, S. Pellegrinelli, N. Pedrocchi, A software tool for human-robot shared-workspace collaboration with task precedence constraints, Robotics and Computer-Integrated Manufacturing 67 (2021) 102051.

[4] L. Yao, W. Yang, W. Huang, A data augmentation method for human action recognition using dense joint motion images, Applied Soft Computing 97 (2020) 106713.

[5] E. P. Ijjina, C. K. Mohan, Hybrid deep neural network model for human action recognition, Applied soft computing 46 (2016) 936–952.

[6] R. Luo, D. Berenson, A framework for unsupervised online human reaching motion recognition and early prediction, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 2426–2433.

[7] H. Ding, G. Reißig, K. Wijaya, D. Bortot, K. Bengler, O. Stursberg, Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 5875–5880.

[8] Q. Li, Z. Zhang, Y. You, Y. Mu, C. Feng, Data driven models for human motion prediction in human-robot collaboration, IEEE Access 8 (2020) 227690–227702.

[9] T. Petković, D. Puljiz, I. Marković, B. Hein, Human intention estimation based on hidden markov model motion validation for safe flexible robotized warehouses, Robotics and Computer-Integrated Manufacturing 57 (2019) 182–196.

[10] T. Petković, J. Hvězda, T. Rybeckỳ, I. Marković, M. Kulich, L. Přeučil, I. Petrović, Human motion prediction framework for safe flexible robotized warehouses, in: IEEE International Conference on Robotics and Automation (ICRA2019), Long-term Human Motion Prediction Workshop, 2019.

[11] J. Mainprice, R. Hayne, D. Berenson, Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 885–892.

[12] T. Petković, J. Hvězda, T. Rybeckỳ, I. Marković, M. Kulich, L. Přeučil, I. Petrović, Human intention recognition for human aware planning in integrated warehouse systems (2020) 1–6.

[13] P. Schydlo, M. Rakovic, L. Jamone, J. Santos-Victor, Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–6.

[14] C.-M. Huang, S. Andrist, A. Sauppé, B. Mutlu, Using gaze patterns to predict task intent in collaboration, Frontiers in Psychology 6 (July) (2015) 1–12. doi:10.3389/fpsyg.2015.01049.

[15] L. Shi, C. Copot, S. Vanlanduit, What are you looking at? detecting human intention in gaze based human-robot interaction, arXiv preprint arXiv:1909.07953 (2019).

[16] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, J. Liu, Online human action detection using joint classification-regression recurrent neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 203–220.

[17] P. Kratzer, N. B. Midlagajni, M. Toussaint, J. Mainprice, Anticipating human intention for full-body motion prediction in object grasping and placing tasks, in: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), IEEE, 2020, pp. 1157–1163.

[18] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2117–2126.

[19] R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu, G. Bebis, Understanding human intentions via hidden markov models in autonomous mobile robots, Proceedings of the 3rd international conference on Human robot interaction - HRI '08 (2008) 367doi:10.1145/1349822.1349870.

[20] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, T. Darrell, Hidden conditional random fields for gesture recognition, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, IEEE, 2006, pp. 1521–1527.

[21] C. Dai, X. Liu, J. Lai, Human action recognition using two-stream attention based lstm networks, Applied soft computing 86 (2020) 105820.

[22] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran, G. Sannino, V. H. C. de Albuquerque, et al., Human action recognition using attention based lstm network with dilated cnn features, Future Generation Computer Systems 125 (2021) 820–830.

[23] Z. Zhang, Z. Lv, C. Gan, Q. Zhu, Human action recognition using convolutional lstm and fully-connected lstm with different attentions, Neurocomputing 410 (2020) 304–316.

[24] D. de Ridder, A. Hoekstra, R. P. Duin, Feature extraction in shared weights neural networks, in: Proceedings of the Second Annual Conference of the Advanced School for Computing and imaging, ASCI, Citeseer, 1996, pp. 289–294.

[25] A. Rozantsev, M. Salzmann, P. Fua, Beyond sharing weights for deep domain adaptation, IEEE transactions on pattern analysis and machine intelligence 41 (4) (2018) 801–814.

[26] M. Liu, J. R. Kitchin, Singlenn: modified behler–parrinello neural network with shared weights for atomistic simulations with transferability, The Journal of Physical Chemistry C 124 (32) (2020) 17811–17818.

[27] L. Shi, C. Copot, S. Vanlanduit, Gazeemd: Detecting visual intention in gaze-based human-robot interaction, Robotics 10 (2) (2021) 68.

[28] T. Bader, M. Vogelgesang, E. Klaus, Multimodal integration of natural gaze behavior for intention recognition during object manipulation, Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI '09 (2009) 199doi:10.1145/1647314.1647350.

[29] A. Buerkle, W. Eaton, N. Lohse, T. Bamber, P. Ferreira, Eeg based arm movement intention recognition towards enhanced safety in symbiotic human-robot collaboration, Robotics and Computer-Integrated Manufacturing 70 (2021) 102137.

[30] S. Jiang, Q. Gao, H. Liu, P. B. Shull, A novel, co-located emg-fmg-sensing wearable armband for hand gesture recognition, Sensors and Actuators A: Physical 301 (2020) 111738.

[31] M. Val-Calvo, J. R. Álvarez-Sánchez, J. M. Ferrández-Vicente, E. Fernández, Affective robot story-telling human-robot interaction: exploratory real-time emotion estimation analysis using facial expressions and physiological signals, IEEE Access 8 (2020) 134051–134066.

[32] D. Osokin, Real-time 2d multi-person pose estimation on cpu: Lightweight openpose, arXiv preprint arXiv:1811.12004 (2018).

[33] H. C. Ravichandar, A. Kumar, A. Dani, Gaze and motion information fusion for human intention inference, International Journal of Intelligent Robotics and Applications 2 (2) (2018) 136–148.

[34] P. Kratzer, S. Bihlmaier, N. Balachandra Midlagajni, R. Prakash, M. Toussaint, J. Mainprice, Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze, IEEE Robotics and Automation Letters (RAL) (2020).

[35] D. Trombetta, G. Rotithor, I. Salehi, A. P. Dani, Variable structure human intention estimator with mobility and vision constraints as model selection criteria, Mechatronics 76 (2021) 102570.

[36] A. P. Dani, I. Salehi, G. Rotithor, D. Trombetta, H. Ravichandar, Human-in-the-loop robot control for human-robot collaboration: Human intention estimation and safe trajectory tracking control for collaborative tasks, IEEE Control Systems Magazine 40 (6) (2020) 29–56.

[37] S. Pellegrini, A. Ess, K. Schindler, L. Van Gool, You'll never walk alone: Modeling social behavior for multi-target tracking, in: IEEE International Conference on Computer Vision (ICCV), 2009, pp. 261–268.

[38] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.

[39] A. Lerner, Y. Chrysanthou, D. Lischinski, Crowds by example, in: Computer graphics forum, Vol. 26, Wiley Online Library, 2007, pp. 655–664.

[40] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, K. O. Arras, Human motion trajectory prediction: A survey, The International Journal of Robotics Research 39 (8) (2020) 895–935.

[41] W. Mao, M. Liu, M. Salzmann, H. Li, Multi-level motion attention for human motion prediction, International Journal of Computer Vision (2021) 1–23.

[42] Carnegie mellon motion capture database, http://mocap.cs.cmu.edu.
URL `http://mocap.cs.cmu.edu`

[43] L. Sigal, M. J. Black, Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion, Brown Univertsity TR 120 (2) (2006).

[44] V. Bloom, D. Makris, V. Argyriou, G3d: A gaming action dataset and real time action recognition evaluation framework, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2012.

[45] P. Kratzer, M. Toussaint, J. Mainprice, Prediction of human full-body movements with motion optimization and recurrent neural networks, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020.

[46] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[47] T. Petković, L. Petrović, I. Marković, I. Petrović, Ensemble of lstms and feature selection for human action prediction, arXiv preprint arXiv:2101.05645 (2021).

[48] J. Bridle, Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters, Advances in neural information processing systems 2 (1989).

[49] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536.

[50] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: Human trajectory prediction in crowded spaces, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 961–971.

[51] M. A. Hall, et al., Correlation-based feature selection for machine learning (1999).

[52] Y. Liu, Y. Mu, K. Chen, Y. Li, J. Guo, Daily activity feature selection in smart homes based on pearson correlation coefficient, Neural Processing Letters 51 (2) (2020) 1771–1787.

[53] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 856–863.

[54] J. Chen, Z. Wu, J. Zhang, Driver identification based on hidden feature extraction by using adaptive nonnegativity-constrained autoencoder, Applied Soft Computing 74 (2019) 1–9.

[55] Q. Meng, D. Catchpoole, D. Skillicom, P. J. Kennedy, Relational autoencoder for feature extraction, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 364–371.

[56] K. Han, Y. Wang, C. Zhang, C. Li, C. Xu, Autoencoder inspired unsupervised feature selection, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 2941–2945.

[57] M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula, Autoencoder-based feature learning for cyber security applications, in: 2017 International joint conference on neural networks (IJCNN), IEEE, 2017, pp. 3854–3861.

[58] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.

[59] Y. Cai, S. Hackett, F. Alber, Interactive indoor localization on helmet, in: International Conference on Applied Human Factors and Ergonomics, Springer, 2020, pp. 544–551.

[60] W. Wei, K. Kurita, J. Kuang, A. Gao, Real-time 3d arm motion tracking using the 6-axis imu sensor of a smartwatch, in: 2021 IEEE 17th International Conference on Wearable and Implantable Body Sensor Networks (BSN), IEEE, 2021, pp. 1–4.

[61] M. C. F. E. Matt J, Absolute orientation - horn's method (2021).
URL `https://www.mathworks.com/matlabcentral/fileexchange/26186-absolute-orientation-horn-s-method`

[62] B. K. Horn, Closed-form solution of absolute orientation using unit quaternions, Josa a 4 (4) (1987) 629–642.

[63] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[64] P. Sedgwick, Greenwood, Understanding the hawthorne effect, British Medical Journal 351 (2015).