Int J Comput Vis
DOI 10.1007/s11263-017-0992-z

Prediction of Manipulation Actions

Cornelia Fermüller[1] · **Fang Wang**[2] · **Yezhou Yang**[6] · **Konstantinos Zampogiannis**[1] · **Yi Zhang**[1] · **Francisco Barranco**[3] · **Michael Pfeiffer**[4,5]

Received: 16 March 2016 / Accepted: 25 January 2017

Abstract By looking at a person's hands, one can often tell what the person is going to do next, how his/her hands are moving and where they will be, because an actor's intentions shape his/her movement kinematics during action execution. Similarly, active systems with real-time constraints must not simply rely on passive video-segment classification, but they have to continuously update their estimates and predict future actions. In this paper, we study the prediction of dexterous actions. We recorded videos of subjects performing different manipulation actions on the same object, such as "squeezing", "flipping", "washing", "wiping" and "scratching" with a sponge. In psychophysical experiments, we evaluated human observers' skills in predicting actions from video sequences of different length, depicting the hand movement in the preparation and execution of actions before and after contact with the object. We then developed a recurrent neural network based method for action prediction using as input image patches around the hand. We also used the same formalism to predict the forces on the finger tips using for training synchronized video and force data streams. Evaluations on two new datasets show that our system closely matches human performance in the recognition task, and demonstrate the ability of our algorithms to predict in real time what and how a dexterous action is performed.

Keywords Online action recognition · Hand motions · Forces on the hand · Action prediction

Communicated by Deva Ramanan and Cordelia Schmid.

✉ Cornelia Fermüller
fer@cfar.umd.edu

[1] University of Maryland, College Park, MD 20742, USA

[2] College of Engineering and Computer Science (CECS), Australian National University, Canberra, Australia

[3] University of Granada, Granada, Spain

[4] Institute of Neuroinformatics, University of Zurich, Zurich, Switzerland

[5] Bosch center for Artificial Intelligence - Research, 71272 Renningen, Germany

[6] Arizona State University, Tempe, AZ 85281, USA

1 Introduction

Human action and activity understanding has been a topic of great interest in Computer Vision and Robotics in recent years. Many techniques have been developed for recognizing actions and large benchmark datasets have been proposed, with most of them focusing on full-body actions (Mandary et al. 2015; Takano et al. 2015; Schuldt et al. 2004; Li et al. 2010; Moeslund et al. 2006; Turaga et al. 2008). Typically, computational approaches treat action recognition as a classification problem, where the input is a previously segmented video and the output a set of candidate action labels.

However, there is more to action understanding, as demonstrated by biological vision. As we humans observe, we constantly perceive and update our belief about both the observed action and future events. We constantly recognize the ongoing action. But there is even more to it. We can understand the kinematics of the ongoing action, the limbs' future positions and velocities. We also understand the observed actions in terms of our own motor-representations. That is, we are able to interpret others' actions in terms of dynamics and forces and predict the effects of these forces on objects. Similarly, cognitive robots that will assist human partners will need to understand their intended actions at an early stage. If a robot needs to act, it cannot have a long delay in visual processing. It needs to recognize in real-time to plan its actions. A fully functional perception-action loop requires

Published online: 20 February 2017

the robot to *predict*, so it can efficiently allocate future processes. Finally, even vision processes for multimedia tasks may benefit from being predictive. Since interpreting human activity is computationally very complex, the task requires a close interaction of different low-level processes with the higher-level cognitive processes (Aloimonos and Fermüller 2015), with prediction playing an essential role in it.

We can think about the action-perception loop of our cognitive system from the viewpoint of a control system. The sensors take measurements of the human activity. We then apply visual operations on this signal and extract (possibly using additional cognitive processes) useful information for creating the control signal in order to change the state of the cognitive system. Because the processing of the signal takes time, this creates a delay for the control (Doyle and Csete 2011). It is therefore important to compute meaningful information that allows us to predict the future state of the cognitive system. In this work, we are specifically interested in manipulation actions and how visual information of hand movements can be exploited for predicting future actions, so that the control loop delay can be minimized (for an illustration, see Fig. 1).

Hand movements and actions have long been studied in Computer Vision to create systems for applications, such as recognition of sign language (Erol et al. 2007). More recent applications include gesture recognition (Molchanov et al. 2015), visual interfaces (Melax et al. 2013), and automotive interfaces (Ohn-Bar and Trivedi 2014). Different methods model the temporal evolution of actions using formalisms such as Hidden Markov models (Starner et al. 1998), Conditional Random Fields (Wang et al. 2006) and 3d Convolutional Neural Networks (Molchanov et al. 2015). While, in principle, some of these approaches could be used for online prediction, they are always treated as recognition modules. In recent years, a number of works have developed tools for general hand pose estimation and tracking, which can become building blocks for applications involving hand movement recognition. For example, Keskin et al. (2013), building on work on full-body motion (Shotton et al. 2013), develops a learning-based approach using depth contrast features and Random Forest classifiers. Oikonomidis et al. (2011), in a model-based approach, use a 27-degree of freedom model of the hand built from geometric primitives and GPU-accelerated Particle Swarm Optimization. So far, these trackers and pose estimators work well on isolated hands, but most methods still struggle with hands in interaction with objects (Supancic et al. 2015), although there are efforts underway to deal with such situations (Panteleris et al. 2015).

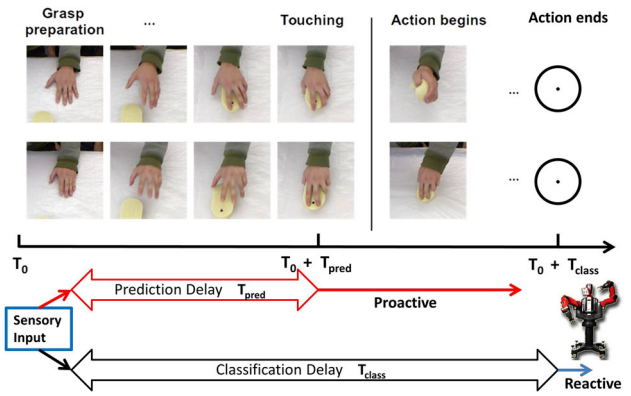


Fig. 1 Two examples demonstrate that early movements are strong indicators of the intended manipulation actions. Inspired by this, our system performs action predictions from early visual cues. Compared to the classification delay, earlier prediction of action significantly reduces the delay in real-time interaction, which is fundamentally important for a proactive system. (*Top row* squeezing a sponge; *bottom row* wiping a table with a sponge)

Inspiration for our work comes from studies in Cognitive Sciences on hand motion. The grasp and the movement kinematics are strongly related to the manipulation action (Jeannerod 1984). It has been shown that an actor's intention shapes his/her movement kinematics during movement execution and, furthermore, observers are sensitive to this information (Ansuini et al. 2015). They can see early differences in visual kinematics and use them to discriminate between movements performed with different intentions. Kinematic studies have looked at such physical differences in movement. For example, Ansuini et al. (2008) found that when subjects grasped a bottle for pouring, the middle and the ring fingers were more extended than when they grasped the bottle with the intent of displacing, throwing, or passing it. Similarly, Crajé et al. (2011) found that subjects placed their thumb and index fingers in higher positions when the bottle was grasped to pour than to lift.

It appears that the visual information in the early phases of the action is often sufficient for observers to understand the intention of action. Starting from this intuition, we (a) conducted a study to evaluate human performance in recognizing manipulation actions; (b) implemented a computational system using state-of-the-art learning algorithms.

The psychophysical experiment was designed to evaluate human performance in recognizing manipulation actions in their early phases. These include: (1) the grasp preparation, which is the phase when the hand moves towards the object and the fingers shape to touch the object; (2) the grasp, when the hand comes in contact with the object to hold it in a stable position; and (3) the early actual action movement of the hand together with the object. Throughout these three phases, observers' judgment of the action becomes more reliable and confident. The study gives us an insight about the difficulty of the task and provides data for evaluating our computational method.

Our computational approach processes the sensory input as a continuous signal and formulates action interpretation

as a continuous update of the prediction of the intended action. This concept is applied to two different tasks. First, from the stream of video input, we continuously predict the identity of the ongoing action. Second, using as input the video stream, we predict the forces on the fingers applied to grasped objects. Next, we provide a motivation for our choice of the two tasks, after which we give an overview of our approach.

The first task is about action prediction from video. We humans are able to update our beliefs about the observed action and predict it before it is completed. This capability is essential to be pro-active and react to the actions of others. Robots that interact with humans also need this capability. Predicting future actions of their counterpart allows them to allocate computational resources for their own reaction appropriately. For example, if a person is passing a cup to the robot, the robot has to understand what is happening well before the action is completed, so it can prepare the appropriate action to receive it. Furthermore, vision processes have to be initiated and possibly tuned with *predicted* information, so the cup can be detected at the correct location, its pose estimated, and possibly other task-specific processes performed (for example, the content of the cup may need to be recognized).

The second task is about predicting the tactile signal of the intended action. Findings of neuroscience on the mirror neuron system (Gallese and Goldman 1998; Rizzolatti et al. 2001) provide evidence for a close relationship between mechanisms of action and perception in primates. Humans develop haptic perception through interaction with objects and learn to relate haptic with visual perception. Furthermore, they develop the capability of hallucinating the haptic stimulus when seeing hands in certain configurations interacting with objects (Tiest and Kappers 2014). This capability of hallucinating force patterns from visual input is essential for a more detailed analysis of the interaction with the physical world. It can be used to *reason about the current interaction* between the hand and the object, and to *predict the action consequences* driven by the estimated force pattern.

Furthermore, by associating vision with forces, we expect to obtain *better computational action recognition* modules. Intuitively, the force vectors, whose dimensions are much lower than the visual descriptors, should provide useful compact information for classification, especially when the training data is not large. A first experiment, presented in Sect. 6.3.3, confirms this idea.

Most importantly, the force patterns may be used in robot learning. A popular paradigm in Robotics is imitation learning or learning from demonstration (Argall et al. 2009), where the robot learns from examples provided by a demonstrator. If the forces can be predicted from images, then the force profiles, together with the positional information, can be used to teach the robot with video only. Many researchers are trying to teach robots actions and skills that involve forces, e.g. wiping a kitchen table (Gams et al. 2010), pull and flip tasks (Kober et al. 2000), ironing or opening a door (Kormushev et al. 2011). These approaches rely on haptic devices or force and torque sensors on the robot to obtain the force profiles for the robot to learn the task. If we can predict the forces exerted by the human demonstrator, the demonstration could become visual only. This would allow us to teach robots force interaction tasks much more efficiently.

In order to solve the above two tasks, we take advantage of new developments in machine learning. Specifically, we build on the recent success of recurrent neural networks (RNNs) in conjunction with visual features from pre-trained convolutional neural networks (CNNs) and training from a limited number of weakly annotated data. For the first task, we use an RNN to recognize the ongoing action from video input. A camera records videos of humans performing a number of manipulation actions on different objects. For example, they 'drink' from a cup, 'pour' from it, 'pound', 'shake', and 'move' it; or they 'squeeze' a sponge, 'flip' it, 'wash', 'wipe', and 'scratch' with it. Our system extracts patches around the hands, and feeds these patches to an RNN, which was trained offline to predict in real-time the ongoing action. For the second task, we collected videos of actions and synchronized streams of force measurements on the hand, and we used this data to train an RNN to predict the forces, using only the segmented hand patches in video input.

The two networks were evaluated in an online real-time system and as standard classification and regression algorithms. We also compared our visual prediction against human performance and evaluated the classification performance on a state-of-the-art dataset (the 50 Salads dataset).

The main contributions of the paper are: (1) we present the first computational study on the prediction of observed dexterous actions (2) we demonstrate an implementation for predicting intended dexterous actions from videos; *3)* we present a method for estimating tactile signals from visual input without considering a model of the object; (4) we provide new datasets that serve as test-beds for the aforementioned tasks.

2 Related Work

We will focus our review on studies along the following concepts: the idea of prediction, including prediction of intention and future events (a), prediction beyond appearance (b), prediction of contact forces on hands (c), work on hand actions (d), manipulation datasets (e), and action classification as a

continuous process using various kinds of techniques and different kinds of inputs (f).

2.1 Prediction of Action Intention and Future Events

A small number of works in Computer Vision have aimed to predict intended action from visual input. For example, Joo et al. (2014) use a ranking SVM to predict the persuasive motivation (or the intention) of the photographer who captured an image. Pirsiavash et al. (2014) seek to infer the motivation of the person in the image by mining knowledge stored in a large corpus using natural language processing techniques. Yang et al. (2015) propose that the grasp type, which is recognized in single images using CNNs, reveals the general category of a person's intended action. In Koppula and Saxena (2016), a temporal Conditional Random Field model is used to infer anticipated human activities by taking into consideration object affordances. Other works attempt to predict events in the future. For example, Kitani et al. (2012) use concept detectors to predict future trajectories in a surveillance videos. Fouhey and Zitnick (2014) learn from sequences of abstract images the relative motion of objects observed in single images. Walker et al. (2014) employ visual mid-level elements to learn from videos how to predict possible object trajectories in single images. More recently, Vondrick et al. (2016) learn, using CNN feature representations, how to predict from one frame in the video the actions and objects in a future frame. Our study is also about prediction of future events using neural networks. But while the above studies attempt to learn abstract concepts for reasoning in a passive setting, our goal is to perform online prediction of specific actions from video of the recent past.

2.2 Physics Beyond Appearance

Many recent approaches in Robotics and Computer Vision aim to infer physical properties beyond appearance models from visual inputs. Xie et al. (2013) propose that implicit information, such as functional objects, can be inferred from video. Zhu et al. (2015) takes a task-oriented viewpoint and models objects using a simulation engine. The general idea of associating images with forces has previously been used for object manipulation. The technique is called vision-based force measurement and refers to the estimation of forces according to the observed deformations of an object (Greminger and Nelson 2004). Along this idea, Aviles et al. (2014) recently proposed a method using an RNN for the classification of forces due to tissue deformation in robotic assisted surgery.

2.3 Inference of Manipulation Forces

The first work in the Computer Vision literature to simulate contact forces during hand-object interactions is Pham et al. (2015). Using as input RGB data, a model-based tracker estimates the poses of the hand and a known object, from which then the contact points and the motion trajectory are derived. Next, the minimal contact forces (nominal forces) explaining the kinematic observations are computed from the Newton–Euler dynamics by solving a conic optimization. Humans typically apply more than the minimal forces. These additional forces are learned using a neural network on data collected from subjects, where the force sensors are attached to the object. Another approach on contact force simulation is due to Rogez et al. (2015). The authors segment the hand from RGBD data in single egocentric views and classify the pose into 71 functional grasp categories as proposed in Liu et al. (2014). Classified poses are matched to a library of graphically created hand poses and these poses are associated with force vectors normal to the meshes at contact points. Thus, the forces on the observed hand are obtained by finding the closest matching synthetic model. Both of these prior approaches derive the forces using model-based approaches. The forces are computed from the contact points, the shape of the hand, and dynamic observations. Furthermore, both use RGBD data, while ours is an end-to-end learning approach using as input only images.

2.4 Dexterous Actions

The robotics community has been studying perception and control problems of dexterous actions for decades (Shimoga 1996). Some works have studied grasping taxonomies (Cutkosky 1989; Feix et al. 2009), how to recognize grasp types (Rogez et al. 2015) and how to encode and represent human hand motion (Romero et al. 2013). Pieropan et al. (2013) proposed a representation of objects in terms of their interaction with human hands. Real-time visual trackers (Oikonomidis et al. 2011) were developed, facilitating computational research with hands. Recently, several learning based systems were reported that infer contact points or how to grasp an object from its appearance (Saxena et al. 2008; Lenz et al. 2015).

2.5 Manipulation Datasets

A number of object manipulation datasets have been created, many of them recorded with wearable cameras providing egocentric views. For example, the Yale grasping dataset (Bullock et al. 2015) contains wide-angle head-mounted camera videos recorded from four people during regular activities with images tagged with the hand grasp (of 33 classes). Similarly, the UT Grasp dataset (Cai et al. 2015)

contains head-mounted camera video of people grasping objects on a table, and was tagged with grasps (of 17 classes). The GTEA set (Fathi et al. 2011) has egocentric videos of household activities with the objects annotated. Other datasets have egocentric RGB-D videos. The UCI-EGO (Rogez et al. 2014) features object manipulation scenes with annotation of the 3D hand poses, and the GUN-71 (Rogez et al. 2015) features subjects grasping objects, where care was taken to have the same amount of data for each of the 71 grasp types. Our datasets, in contrast, are taken from the third-person viewpoint. While they have less variation in the visual setting than most of the above datasets, they focus on the dynamic aspects of different actions performed on the same objects.

2.6 Action Recognition as an Online Process

Action recognition has been extensively studied. However, few of the proposed methods treat action recognition as a continuous (in the online sense) process; typically, action classification is performed on *whole* action sequences (Schuldt et al. 2004; Ijina and Mohan 2014). Recent works include building robust action models based on MoCap data (Wang et al. 2014) or using CNNs for large-scale video classification (Karpathy et al. 2014; Simonyan and Zisserman 2014). Most methods that take into account action dynamics usually operate under a stochastic process formulation, e.g., by using Hidden Markov Models (Lv and Nevatia 2006) or semi-Markov models (Shi et al. 2011). HMMs can model relations between consecutive image frames, but they cannot be applied to high-dimensional feature vectors. In Fanello et al. (2013), the authors propose an online action recognition method by means of SVM classification of sparsely coded features on a sliding temporal window. Most of the above methods assume only short-time dependencies between frames, make restrictive assumptions about the Markovian order of the underlying processs and/or rely on global optimization over the whole sequence.

In recent work, a few studies proposed approaches to recognition of partially observed actions under the headings of *early event detection* or *early action recognition*. Ryoo (2011) creates a representation that encodes how histograms of spatio-temporal features change over time. In a probabilistic model, the histograms are modeled with Gaussian distributions and MAP estimation over all subsequences is used to recognize the ongoing activity. A second approach in the paper models the sequential structure in the changing histogram representation and matches subsequences of the video using dynamic programming. Both approaches were evaluated on full body action sequences. In Ryoo and Matthies (2013), images are represented by spatio-temporal features and histograms of optical flow and a hierarchical structure of video-subsegments is used to detect partial action sequences in first-person videos. Ryoo et al. (2015) perform early recognition of activities in first person-videos by capturing special sub-sequences characteristic for the onset of the main activity. Hoai and De la Torre (2014) propose a maximum-margin framework (a variant of SVM) to train visual detectors to recognize partial events. The classifier is trained with all the video sub-sequences of different length. To enforce the sequential nature of the events, additional constraints on the score function of the classifier are enforced (e.g., the score has to increase as more frames are matched). The technique was demonstrated in multiple applications, including detection of facial expressions, hand gestures, and activities.

The main learning tools used here, the RNN and the Long Short Term Memory (LSTM) model, were recently popularized in language processing and have been used for translating videos to language (Venugopalan et al. 2014), image description generation (Donahue et al. 2015), object recognition (Visin et al. 2015), and the estimation of object motion (Fragkiadaki et al. 2015). RNNs were also used for action recognition (Ng et al. 2015) to learn dynamic changes within the action. The aforementioned paper still performs whole video classification by using average pooling and does not consider the use of RNNs for prediction. In a very recent work, however, Ma et al. (2016) train a LSTM using novel ranking losses for early activity detection. Our contribution regarding action recognition is not that we introduce a new technique. We use an existing method (LSTM) and demonstrate it in an online prediction system. The system keeps predicting and considers the prediction reliable when the predicted label converges (i.e. stays the same over a number of frames). Furthermore, the subject of our study is novel. The previous approaches consider the classical full body action problem. Here, our emphasis is specifically on the hand motion, without considering other information such as the objects involved.

3 Our Approach

In this section, we first review the basics of Recurrent Neural Networks (RNNs) and the Long Short Term Memory (LSTM) model and then describe the specific algorithms for prediction of actions and forces used in our approach.

3.1 Recurrent Neural Networks

Recurrent Neural Networks have long been used for modeling temporal sequences. The recurrent connections are feedback loops in the unfolded network and, because of these connections, RNNs are suitable for modeling time series with strong nonlinear dynamics and long time correlations.

Given a sequence $x = \{x_1, x_2, \ldots, x_T\}$, an RNN computes a sequence of hidden states $h = \{h_1, h_2, \ldots, h_T\}$ and outputs $y = \{y_1, y_2, \ldots, y_T\}$, for each $t \in [1, T]$, as follows:

$$h_t = \mathscr{H}(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = \mathscr{O}(W_{ho}h_t + b_o), \quad (2)$$

where W_{ih}, W_{hh}, W_{ho} denote weight matrices, b_h, b_o denote the biases, and $\mathscr{H}(\cdot)$ and $\mathscr{O}(\cdot)$ are the activation functions of the hidden layer and the output layer, respectively. Typically, the activation functions are defined as the hyperbolic tangent function.

The traditional RNN is hard to train due to the so called *vanishing gradient* problem, i.e. the weight updates computed via error back-propagation through time may become very small. The Long Short Term Memory model (Hochreiter and Schmidhuber 1997) has been proposed as a solution to overcome this problem. The LSTM architecture uses several gates to control the information flow that passed in or out of the memory cell. This mechanism alleviates the vanishing gradient problem in back-propagation over time and makes the optimization procedure more robust.

Specifically, the LSTM architecture includes an input gate i_t, a forget gate f_t, an output gate o_t, and a memory cell c_t, as shown in Fig. 2. These components, including the gates and the memory cell, are updated as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (5)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$h_t = o_t \circ \tanh(c_t), \quad (7)$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\circ$ denotes the Hadamard product. In this architecture, the input gate i_t controls the portion of new information that is allowed to flow into the memory cell. The forget gate f_t controls the portion of information retained in the memory cell and the output gate o_t determines how much the new information affects the output activation. All these three gates use the sigmoid activation functions, while the cell state c_t and the hidden state h_t use the hyperbolic tangent activation functions. These mechanisms allow the LSTM network to learn temporal dynamics with long time constants.

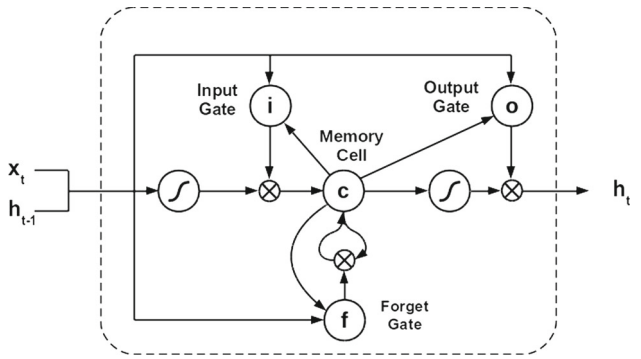


Fig. 2 A diagram of a LSTM memory cell [adapted from Graves et al. (2013)]

3.2 RNN for Action Prediction

In this section, we describe our proposed model for prediction of manipulation actions, in which a human operator manipulates an object using one hand. Given a video sequence of a manipulation action, the goal is to generate a sequence of belief distributions over the predicted actions while watching the video. Instead of assigning an action label to the whole sequence, we continuously update our prediction as frames of the video are processed.

3.2.1 Visual Representation

The visual information most essential for manipulation actions is due to the pose and movement of the hand, while the body movements are less important. Therefore, we first track the hand, using a mean-shift based tracker (Bradski 1998), and use cropped image patches centered on the hand as our inputs. In order to generate visual representations of these image patches, we project each patch through a pre-trained CNN model to get the feature vectors (shown in Fig. 3). We use the VGG 16-layer model (Simonyan and Zisserman 2014) to process the visual features; more details will be discussed in Sect. 6.1.

3.2.2 Action Prediction

In our model, the LSTM is trained using as input a sequence of feature vectors $x = \{x_1, x_2, \cdots, x_T\}$ and the action labels $y \in [1, N]$. The hidden states and the memory cell values are updated according to Eqs. (3)–(7). Logistic regression is then used to map the hidden states to the label space as follows:

$$P(Y = i|h_t, W_u, b_u) = \text{softmax}_i(W_u h_t + b_u), \ i \in [1, N]. \quad (8)$$

The predicted action label of each frame can then be obtained using:

$$\hat{y}_t = \text{argmax}_{i=1}^{N} P(Y = i|h_t, W_u, b_u). \quad (9)$$

Model Learning For each training sample, we define the loss function as the normalized negative log-likelihood over the whole sequence as:

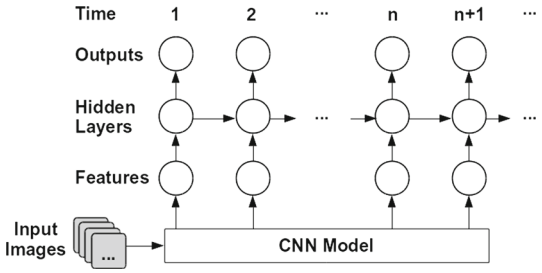


Fig. 3 The flowchart of the action prediction model, where the LSTM model is unfolded over time

$$l_0(x, y, W, b) = -\frac{1}{T}\sum_{t=0}^{T}\log(P(Y = y|x_t, W, b)), \tag{10}$$

where W and b denote the weight matrix and the bias terms. Then, following the common approach, we can train the model by minimizing the loss function over the dataset $\mathscr{D}$. The loss function for action prediction is defined as:

$$l_{prediction}(\mathscr{D}, W, b) = \sum_{j=0}^{|\mathscr{D}|} l_0(x^{(j)}, y^{(j)}, W, b), \tag{11}$$

where $x^{(j)}$ and $y^{(j)}$ denote the j-th sample value and corresponding label in the training set. The parameters W and b can be learned using the stochastic gradient descent algorithm.

Since we aim for ongoing prediction rather than classification of the whole sequence, we do not perform pooling over the sequences. Each prediction is based only on the current frame and the current hidden state, which implicitly encodes information about the history. In practice, we achieve learning by performing back-propagation at each frame.

3.3 RNN for Prediction of Forces at the Fingers

We use a model similar to the one above to predict the forces on the fingers from visual input. Given video sequences of actions, as well as simultaneously recorded sequences of force measurements (see Sect. 4.1), we reformulate the LSTM model such that it predicts force estimates as close as possible to the ground truth values.

As before, we use as input to the LSTM, features from pre-trained CNNs applied to image patches. In addition, the force measurements $v = \{v_1, v_2, \ldots, v_T\}$, $v_t \in R^M$, are used as target values, where M is the number of force sensors attached to the hand. The forces are then estimated as:

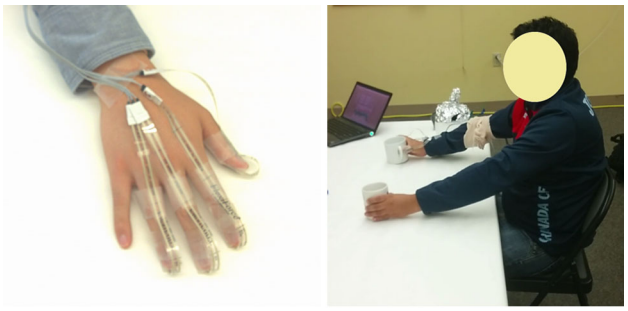
$$\hat{v_t} = W_v h_t + b_v. \tag{12}$$

To train the force estimation model, we define the loss function as the least squares distance between the estimated value and the ground truth and minimize it over the training set using stochastic gradient descent as:

$$l_1(x, v, W, b) = \frac{1}{T}\sum_{t=0}^{T}\|\hat{v_t} - v_t\|_2^2, \tag{13}$$

$$l_{regression}(\mathscr{D}, W, b) = \sum_{j=0}^{|\mathscr{D}|} l_1(x^{(j)}, v^{(j)}, W, b), \tag{14}$$

where $x^{(j)}$ and $v^{(j)}$ denote the j-th sample, and W and b denote the model parameters.



(a) **(b)**

Fig. 4 Illustration of the force-sensing device. **a** The sensors attached to four fingers; **b** The scenario of data collection

4 Data Collection

4.1 A Device for Capturing Finger Forces During Manipulation Actions

We made a force sensing device with four force sensors attached directly to four fingers: the thumb, the pointer, the middle and the ring finger (see Fig. 4a). We omitted the small finger, as the forces on this finger are usually quite small and not consistent across subjects [as found also by Pham et al. (2015)]. We used the piezoresistive force sensors by Tekscan, with a documented accuracy (by the manufacturer) of ±3%. The sensors at the finger tips have a measurement range of 0 to 8.896 N (2 lb), with a round sensing area of 9.53 mm in diameter. The entire sensing area is treated as one single contact point.

The raw sensor outputs are voltages, from which we can calculate the force values perpendicular to the sensor surfaces as:

$$F = 4.448 * \left(C_1 * \frac{V_{out}}{V_{in} - V_{out}} - C_2\right), \tag{15}$$

where V_{out} is the sensor measurement. V_{in}, C_1, and C_2 are fixed constants of the system. To remove environmental noise, we applied notch filtering to the raw data, which

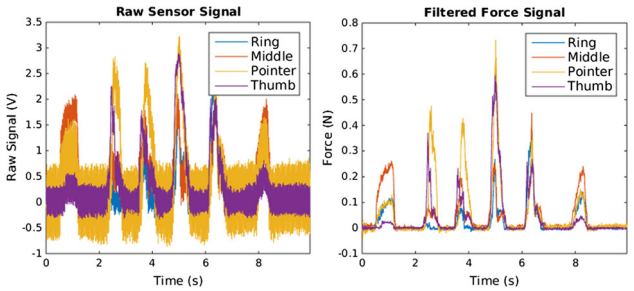


Fig. 5 Example of collected force data. **a** The raw, unfiltered voltage signal from the fingertip force sensors. **b** The filtered force signal from the fingertip sensors

Table 1 Object and action pairs in MAD

Object	Actions
Cup	Drink, pound, shake, move, pour
Stone	Pound, move, play, grind, carve
Sponge	Squeeze, flip, wash, wipe, scratch
Spoon	Scoop, stir, hit, eat, sprinkle
Knife	Cut, chop, poke a hole, peel, spread

gave us clear and smooth force outputs (see Fig. 5). The software, which we designed for the device, will be released as a ROS package, including data recording and force visualization modules.

4.2 Manipulation Datasets

Two datasets were collected. The first dataset contains videos of people performing dexterous actions on various objects. The focus was to have different actions (with significant variation) on the same object. This dataset was used to validate our approach of visual action prediction. The second dataset contains simultaneously recorded video and force data streams, but it has fewer objects. It was used to evaluate our approach of hand force estimation.

4.2.1 Manipulation Action Dataset (MAD)

We asked five subjects to perform a number of actions with five objects, namely *cup*, *stone*, *sponge*, *spoon*, and *knife*. Each object was manipulated in five different actions with five repetitions, resulting in a total of 625 action samples. Table 1 lists all the object and action pairs considered in MAD.

Since our aim was to build a system that can predict the action as early as possible, we wanted to study the prediction performance during different phases in the action. To facilitate such studies, we labeled the time in the videos when the hand establishes contact with the objects, which we call the "touching point."

4.2.2 Hand Actions with Force Dataset (HAF)

To solve the problem of synchronization, we asked subjects to wear on their right hand the force sensing device, leave their left hand bare, and then perform with both hands the same action, with one hand mirroring the other (see Fig. 4b) for the setting). We recorded data from five subjects performing different manipulation actions on four objects, as listed in Table 2. Each action was performed five times, resulting in a total of 500 sample sequences.

5 An Experimental Study with Humans

We were interested in how humans perform in prediction at different phases during the action. Intuitively, we would expect that the hand configuration and motion just before the grasping of the object, when establishing contact, and shortly after the contact point can be very informative of the intended action. Therefore, in order to evaluate how early we can accurately predict, we investigated the prediction performance at certain time offsets with respect to the touching point.

We picked three objects from the MAD dataset for the study, namely *cup*, *sponge* and *spoon*. The prediction accuracy at four different time points was then evaluated: 10 frames before the contact point, exactly at contact, 10, and 25 frames after the contact point. Figure 6 shows the interface subjects used in this study.

Table 2 Object and action pairs in HAF

Object	Actions
Cup	Drink, move, pound, pour, shake
Fork	Eat, poke a hole, pick, scratch, whisk
Knife	Chop, cut, poke a hole, scratch, spread
Sponge	Flip, scratch, squeeze, wash, wipe

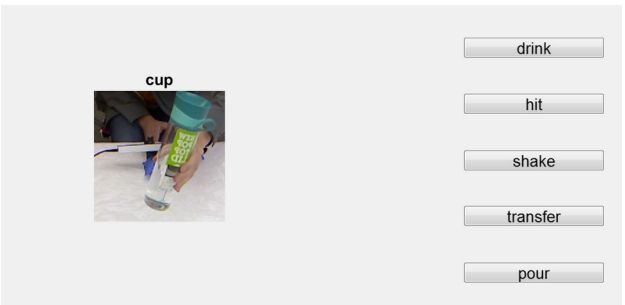


Fig. 6 Interface used in the human study

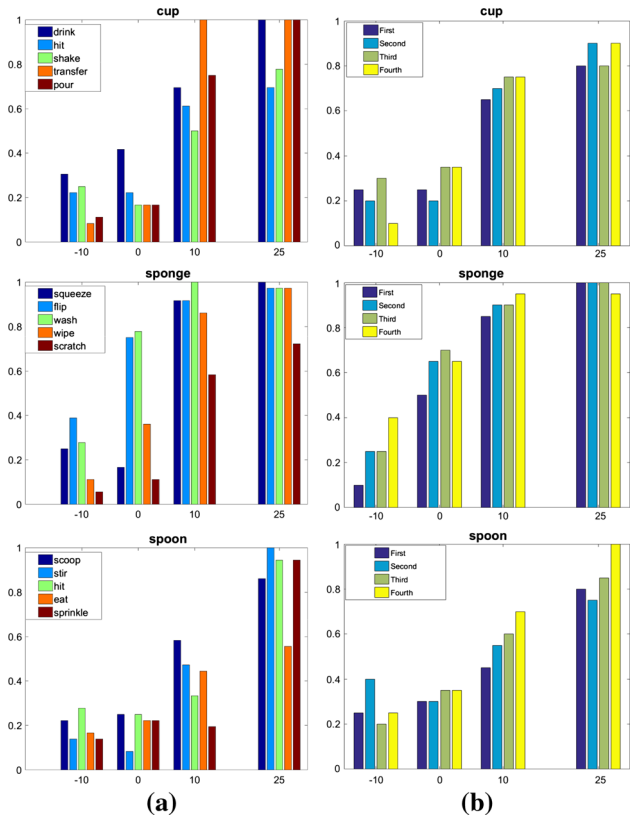


Fig. 7 Human prediction performance. **a** First study (without feedback). Success rate for three objects (cup, sponge, and spoon) for five different actions at four time offsets. **b** Second study (with feedback). Success rate for three objects averaged over five actions over four sets of videos at four offsets

In a first experiment, we asked 18 human subjects to perform the prediction task. For each of the three objects, after a short "training" phase in which all actions were demonstrated at full length, each subject was shown a set of 40 video segments and was asked to identify the currently perceived action. Each segment ended at one of the four time points relative to the contact point described above and was constructed from the same hand patches used in the computational experiments. All actions and all time offsets were equally represented. Figure 7a plots the subjects' average prediction performance for the different objects, actions and time offsets. With five actions per object, 20% accuracy corresponds to chance level. As we can see, the task of judging before and even at contact point, was very difficult and classification was at chance for two of the objects, the spoon and and the cup, and above chance at contact only for the sponge. At 10 frames after contact, human classification becomes better and reaches in average about 75% for the sponge, 60% for the cup, but only 40% for the spoon. At 25 frames subjects' judgment becomes quite good with the sponge going above 95% for four of the five actions, and the other two actions in average at about 85%. We can also see which actions are easily confused. For the cup, 'shake' and 'hit' were still difficult to recognize even after 25 frames. For the spoon, the early phases of movement for most actions appeared similar, and 'eat' was most difficult to identify.

To see whether there is additional distinctive information in the actors' movement and whether subjects can take advantage of it with further learning, we performed a second study. Five participating subjects were shown 4 sets of 40 videos for each object, but this time they were given feedback on which was the correct action. Figure 7b shows the overall success rate for each object and time offset over the four sets. If learning occurs, subjects should improve from the first to the fourth set. The graphs show that there is a bit of learning; its effect is largest for the spoon, where subjects can learn to better distinguish at 10 frames after contact.

6 Experimental Results

The two algorithms have been implemented in a system that runs in real-time on a GPU. This section reports experiments on three datasets. Experiments in Sect. 6.1 evaluate on MAD the prediction performance as an ongoing task and as classical recognition for the whole sequence, and compare the action recognition algorithm against human performance. Section 6.2 evaluates the visual recognition algorithm on a dataset from the literature, and Sect. 6.3 evaluates the force estimation using HAF.

6.1 Hand Action Prediction on MAD

Our approach uses visual features obtained with deep learning, which serve as input to a sequence learning technique.

First, we apply the mean-shift based tracker of Comaniciu et al. (2000) to obtain the locations of the hand. We crop image patches of size 224×224 pixels, centered on the hand. Then our feature vectors are computed by projecting these patches through a convolutional neural network. To be specific, we employ the VGG network (Simonyan and Zisserman 2014) with 16 layers, which has been pre-trained on the ImageNet (Russakovsky et al. 2015). We take the output of layer "fc7" as feature vector (4096 dimensions), which we then use to train a one layer LSTM RNN model for action prediction.

Our RNN has hidden states of 64 dimensions, with all the weight matrices randomly intialized using the normal distribution. We first learn a linear projection to map the 4096 input features to the 64 dimensions of the RNN. We use mini-batches of 10 samples and the adaptive learning rate method to update the parameters. For each mini-batch, the input sequences are aligned with the longest sequence by padding zeros at the end. The training stops after 100 epochs in all the experiments.

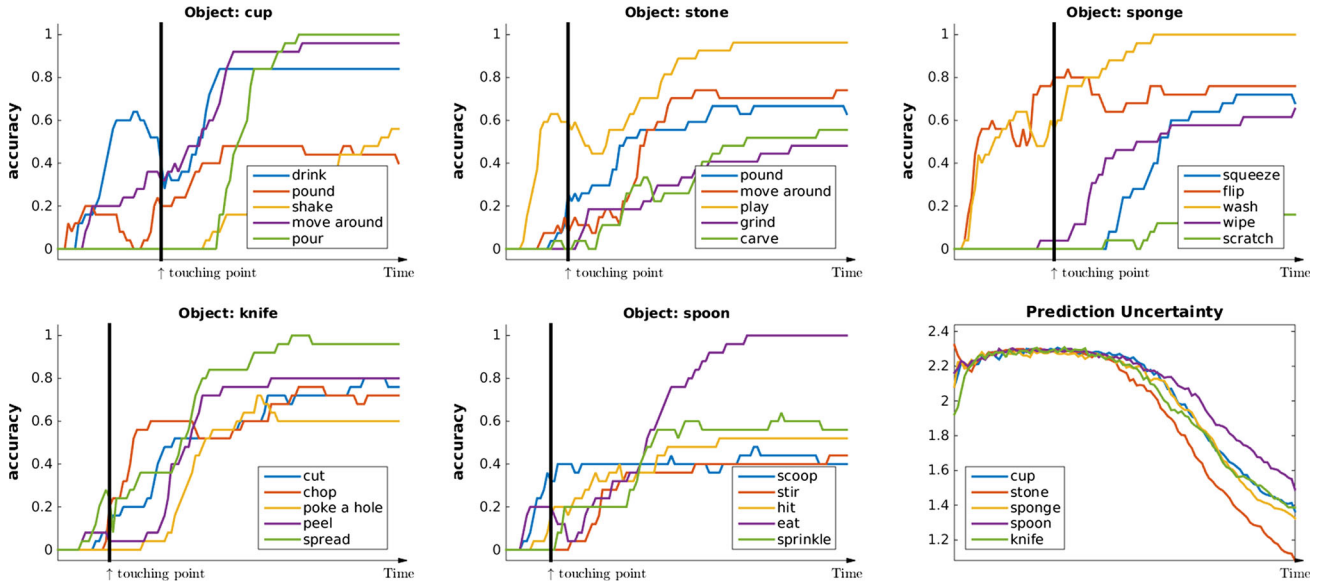


Fig. 8 Prediction accuracies over time for the five different objects, and prediction uncertainty computed from the entropy. The *black vertical bars* show the touching point. For each object we warped and aligned all the sample sequences so that they align at the same touching point. Best viewed in *color*

To evaluate the action prediction performance, we performed leave-one-subject-out cross-validation over the five subjects. Each time, we used the data from one subject for testing and trained the model on the other four subjects. Then all the results were averaged over the five rounds of testing.

6.1.1 On-Going Prediction

Our goal is to understand how the recognition of action improves over time. Thus, we plot the prediction accuracy as a function of time, from the action preparation to the end of the action. Our system performs predictions based on every new incoming frame as the action unfolds.

The first five plots in Fig. 8 show the change in prediction accuracy over time. For a given action video, our system generates for each frame a potential score vector (with one value for each action) to form a score sequence of same length as the input video. Since the actions have different length, we aligned them at the touching points. To be specific, we resampled the sequences before and after the touching points to the same length. For each object, we show the prediction accuracy curves of the five actions.

The vertical bar in each figure indicates the time of the *touching point*. The touching point splits the sequence into two phases: the "preparation" and the "execution". It is interesting to see that for some object-action pairs our system yields high prediction accuracy even before the touching point, e.g. the "cup - drink" and "sponge - wash".

The last plot in Fig. 8 shows the change of prediction uncertainty over time for each of the five objects. This measure was derived from the entropy over the different actions.

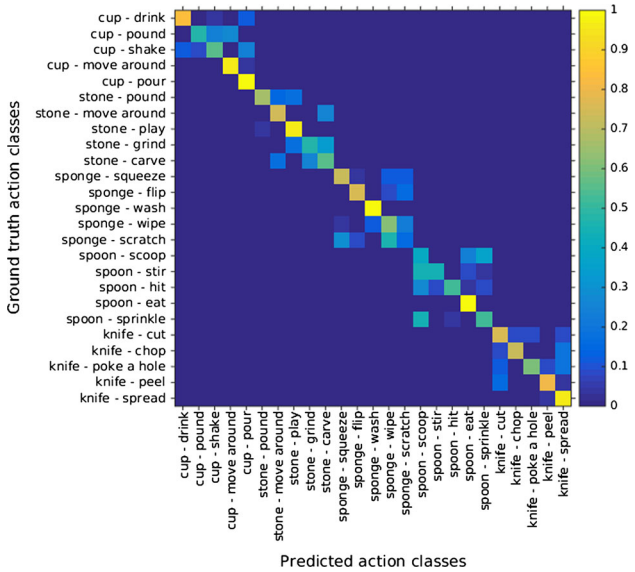


Fig. 9 Confusion matrix of action classification

As can be seen, in all cases, the uncertainty drops rapidly as the prediction accuracy rises along time.

6.1.2 Classification Results

At the end of the action, the on-going prediction task becomes a traditional classification. To allow evaluating our method on classical action recognition, we also computed the classification results for the whole video. The estimate over the sequence was derived as a weighted average over all frames

Table 3 Comparison of classification accuracies on different objects

Object/action	SVM	HMM	LSTM HOG	LSTM VGG16
Cup/drink	79.1	96.0	82.9	92.5
Cup/pound	20.0	81.7	40.0	73.3
Cup/shake	64.3	56.8	32.6	83.3
Cup/move	62.7	53.2	51.9	82.1
Cup/pour	60.0	100.0	80.3	80.8
Stone/pound	26.7	73.3	60.0	73.3
Stone/move	87.8	68.0	90.0	61.4
Stone/play	64.6	97.1	60.5	86.7
Stone/grind	28.3	45.0	60.0	46.7
Stone/carve	43.3	28.5	66.0	39.1
Sponge/squeeze	41.1	81.7	64.3	83.4
Sponge/flip	53.3	91.0	96.0	71.0
Sponge/wash	85.9	84.6	91.1	92.5
Sponge/wipe	46.9	47.5	58.1	46.3
Sponge/scratch	30.0	0.0	43.3	15.0
Spoon/scoop	39.0	27.1	53.6	32.0
Spoon/stir	45.3	30.0	20.0	74.3
Spoon/hit	28.9	20.0	22.4	56.7
Spoon/eat	65.0	79.2	78.1	81.1
Spoon/sprinkle	60.0	25.0	40.5	69.1
Knife/cut	33.5	33.7	49.6	75.3
Knife/chop	0.0	45.0	43.3	72.7
Knife/poke a hole	33.3	20.0	51.0	72.0
Knife/peel	66.3	28.9	90.0	72.5
Knife/spread	38.2	28.3	54.3	74.2
Avg.	48.1	53.7	59.2	68.3

Values are expressed in percentages

using a linear weighting with largest value at the last frame. To be consistent with the above, the classification was performed for each object over the five actions considered.

Figure 9 shows the confusion matrix of the action classification results. One can see that our model achieved high accuracy on various object-action combinations, such as "cup /drink" and "sponge/wash", where the precision exceeds 90%.

We used two traditional classification methods as our baseline: Support Vector Machine (SVM) and Hidden Markov Model (HMM). For the HMM model, we used the mixture of Gaussian assumption and we chose the number of hidden states as five. Since the SVM model doesn't accept input samples of different length, we used a sliding window ($size = 36$) mechanism. We performed the classification over each window and then combined the results using majority voting. For both these baseline methods, we conducted a dimension reduction step to map the input feature vectors to 128 dimensions using PCA. To further explore the efficiency of the LSTM method in predicting actions on our dataset, we also applied the LSTM model using HoG features as input. The average accuracy was found 59.2%, which is 10% higher than the HMM and 23% higher than the SVM, but still significantly lower than our proposed method (Table 3).

It should be noted that this is a novel, challenging dataset with no equivalent publicly available counterparts. Subjects performed the action in unconstrained conditions, and thus there was a lot of variation in their movement, and they performed some of the actions in very similar ways, making them difficult to distinguish, as also our human study confirms.

The results demonstrate that deep learning based continuous recognition of manipulation actions is feasible, providing a promising alternative to traditional methods such as HMM, SVM, and other methods based on hand-crafted features.

6.1.3 Action Prediction at the Point of Contact, Before and After

We next compare the performance of our online algorithm (as evaluated in Sect. 6.1.1) against those of human subjects. Figure 10 summarizes the prediction performance per object

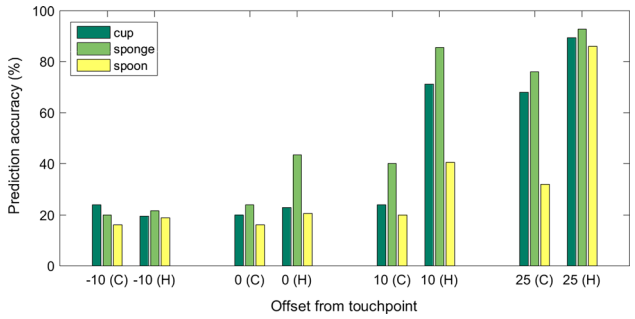


Fig. 10 Comparison of prediction accuracies between our computational method (C) and data from human observers (H). Actions are classified at four different time points before, at, and after the touching point (at $-10, 0, +10, +25$ frames from the touching point). C denotes the learned model, H denotes the psychophysical data)

and time offset. As we can see, our algorithm's performance is not significantly behind that of humans. At ten frames after contact, computer lags behind human performance. However, at 25 frames after the contact point, the gaps between our proposed model and human subjects are fairly small. Our model performs worse on the spoon, but this is likely due to the large variation in the way different people move this object. Our human study already revealed the difficulty in judging spoon actions, but the videos shown to subjects featured less actors than were tested with the algorithm. Considering this, we can conclude that our algorithm is already close to human performance in fast action prediction.

6.2 Action Prediction on Continuous Sequences

To further validate the effectiveness of our approach, we applied our model to the 50 Salads dataset (Stein and McKenna 2013), which has 50 videos featuring 25 subjects preparing two salads each. Each video contains multiple actions, which are labeled at different levels of granularity. To simplify the comparison, we chose the second coarsest level using ten different action labels, as defined in Stein and McKenna (2013).

Two evaluation metrics are used: the frame-wise classification accuracy and the edit score of the generated segmentations. The edit score is defined as $(1 - D(y, \hat{y})) \times 100.0$, where y denotes the ground truth segmentation, $\hat{y}$ denotes the predicted segmentation, and $D(y, \hat{y})$ is the normalized Levenshtein distance of the two sequences. This distance, usually used in string matching, is a function of segment insertions, deletions and substitutions. Thus, higher edit score values indicate more consistency between the predicted sequences and the ground truth.

In this dataset, each video contains sequences of consecutive action and this, not using a presegmentation, makes the task of action prediction much more difficult. We compare to the method proposed in Lea et al. (2016), which

Table 4 Comparison of classification accuracies on the 50 salad dataset

Method	Accuracy	Edit score
ST-CNN + Seg	72.00%	62.06
Our approach	88.50%	50.25

Our method is compared against (Lea et al. 2016)

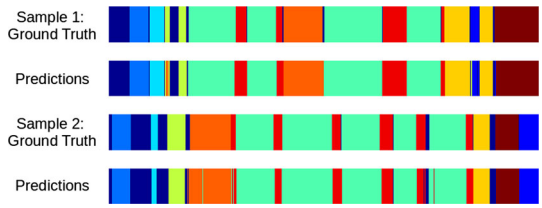


Fig. 11 Samples of action prediction on the 50 salad dataset. Each *color* corresponds to a different class label. The first and third row show the ground truth action classes, while the second and fourth row show the predicted class labels using our method

combines spatial-temporal CNN models with a video segmentation method. In contrast, we do not use any kind of video segmentation in our prediction framework. Benefited by the LSTM model, our method can automatically learn the transitions between different actions and produce accurate predictions over all frames. Referring to Table 4, our method achieved higher performance, raising the frame-wise accuracy from 72.00 to 88.50%. However, our method has a lower edit score, because, without additional temporal constraints, fluctuating predictions around the action boundaries are inevitable. This effect can also be seen in Fig. 11, which shows two prediction examples of consecutive actions. As can also be seen from the figure, in general, the method predicts the actions very well and it does so accurately after a short onset of each new action. In summary, the experiment demonstrates the usefulness of the proposed method for the prediction of manipulation actions and that it is applicable to challenging state-of-the-art datasets.

6.3 Hand Force Estimation on HAF

In the following, we demonstrate the ability of the RNN to predict the forces on the fingers directly from images. We developed an online force estimation system. While watching a person performing actions in front of the camera, our system provides the finger forces in real time. Figure 12 shows one example of online force prediction for the "drink" action. We next describe our training method and then present our results.

6.3.1 Training

The LSTM model (described in Sect. 3.3) is used to estimate the hand forces for each frame. Since people have different

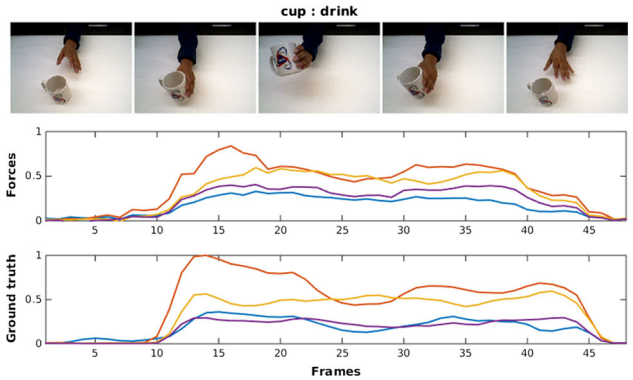


Fig. 12 Illustration of the online force estimation system. The video frames in the *top row* show samples of the action 'drinking from a cup.' The *second row* shows the estimated forces and the *third row* the corresponding ground truth

preferences in performing actions, the absolute force values can vary significantly for the same action. Therefore, we first normalize the force samples, which are used for training, to the range [0, 1]. The visual features in the video frames are obtained the same way as in the action prediction. Our LSTM model has one layer with 128 hidden states. To effectively train the model, we use the adaptive learning rate method for updating the neural network with a batch size of 10 and stop the training at 100 epochs.

6.3.2 Results

We first show examples of our force estimation and then report the average errors. Figure 13 shows six sample results. For each of the samples, the first column shows the ground truth curves, while the second column shows the estimated

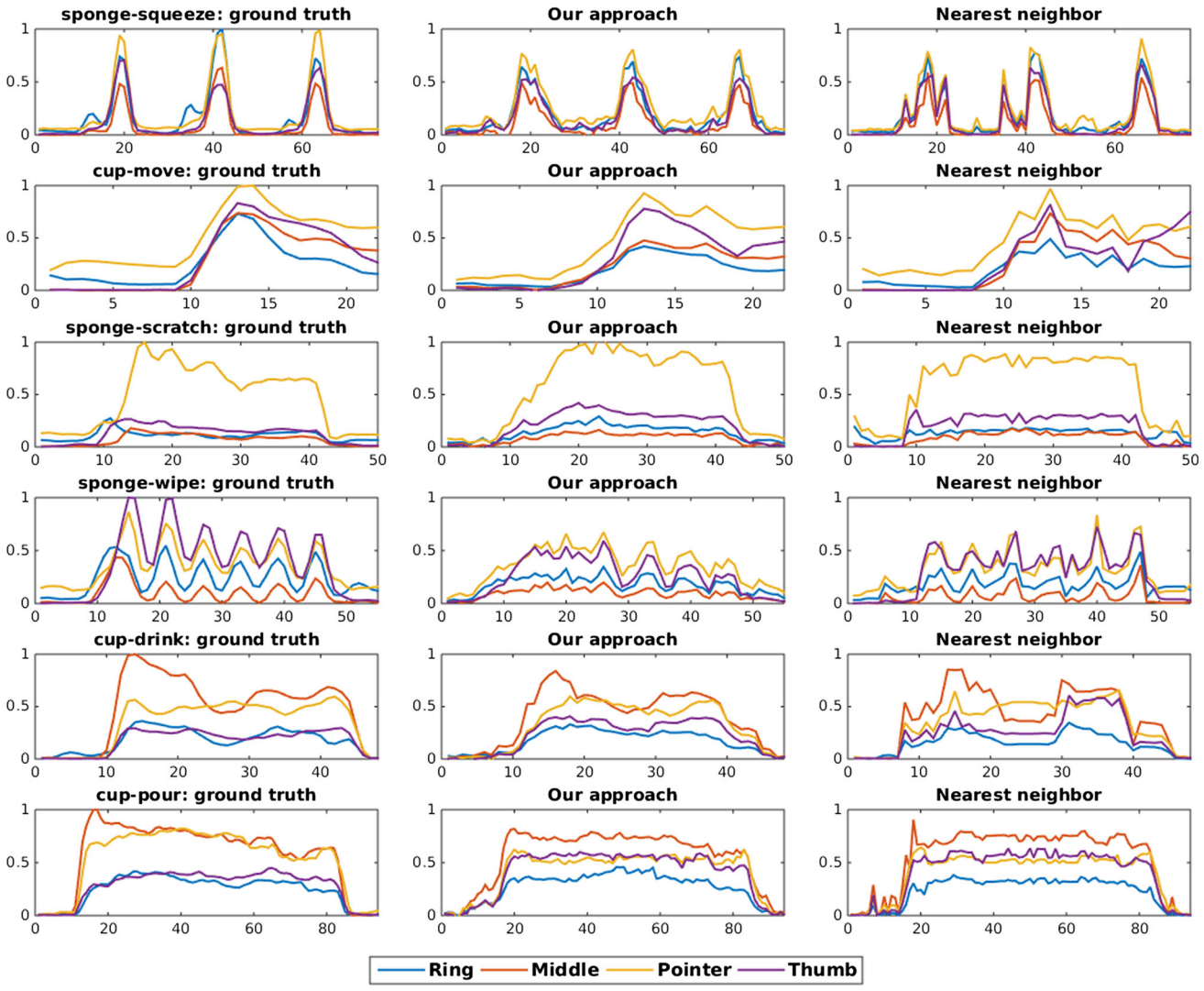


Fig. 13 Samples of force estimation results. The *first column* show the ground truth force curves of six actions, the *second column* show our estimation results using LSTM regressor, and the *third column* illustrate the results of nearest neighbor method for comparison

forces using our approach. To provide a baseline result for this experiment, we generated force curves using nearest neighbor search for each testing frame. To be specific, we used the "fc7" layer of the VGG feature network and reduced the dimensions to 128 using Principal Component Analysis. The third column shows the results of the nearest neighbor method.

From Fig. 13, it can be seen that our system estimates well the overall force patterns for different actions. For example, for the "sponge/squeeze" action, the estimated forces correctly reproduce the three peaks of the real action, and for the "cup/move" action, the output forces predict the much smoother changes. For most cases, the LSTM regression method can recover more accurately the dynamic patterns of the force curves, while the values of the baseline method fluctuate more strongly.

Table 5 provides the average error of the estimated forces for each finger and Table 6 gives the average estimation error for all the actions. The errors are in the range of 0.075–0.155, which demonstrates that the method also has good quantitative prediction and potential for visual force prediction. It can also be seen that our method yields less average errors than the nearest neighbor method in almost all the comparisons.

6.3.3 Why Predict Forces?

One motivation for predicting forces is that the additional data, which we learned through association, may help increase recognition accuracy. There is evidence that humans understand others' actions in terms of their own motor primitives (Gallesse and Goldman 1998; Rizzolatti et al. 2001). However, so far these findings have not been modeled in computational terms.

To evaluate the usefulness of the predicted forces, we applied our force estimation algorithm, which was trained on HAF, to the MAD dataset, to compute the force values. Then, we used the vision data together with the regressed force values as bimodal information to train a network for action prediction. Table 7 shows the results of the prediction accuracy with the bimodal information on different objects. Referring to the table, the overall average accuracy for the combined vision force data (V + F) was 2.2% higher than for vision information only. This first attempt on predicting with bimodal data demonstrates the potential of utilizing visually estimated forces for recognition. Future work will further elaborate on the idea and explore networks (Hoffman et al. 2016) which can be trained from both vision and force at the same time to learn "hallucinate" the forces and predict actions.

As discussed in the introduction, the other advantage is that we will be able to teach robots through video demonstration. If we could predict forces exerted by the human demonstrator and provide the force profile of the task using vision only, this would have a huge impact on the way robots learn force interaction tasks. In future work, we plan to develop and employ sensors that can also measure the tangential forces on the fingers, i.e. the frictions. We also will expand the sensor coverage to the whole hand. With these two improvements, our method could be applied to a range of complicated tasks, such as screwing or assembling.

Table 5 Average errors of estimated force for each finger

Methods	Ring	Middle	Pointer	Thumb
NN	0.117	0.116	0.157	0.148
Ours	0.103	0.098	0.130	0.119

Table 6 Average errors of estimated force for each action

Cup	Drink	Move	Pound	Pour	Shake
NN	0.121	0.145	0.176	0.121	0.152
Ours	0.096	0.122	0.108	0.107	0.110
Fork	**Eat**	**Hole**	**Pick**	**Scratch**	**Whisk**
NN	0.119	0.115	0.078	0.113	0.127
Ours	0.106	0.090	0.075	0.094	0.100
Knife	**Chop**	**Cut**	**Poke**	**Scratch**	**Spread**
NN	0.181	0.167	0.132	0.154	0.132
Ours	0.157	0.155	0.109	0.123	0.110
Sponge	**Flip**	**Scratch**	**Squeeze**	**Wash**	**Wipe**
NN	0.107	0.134	0.126	0.149	0.137
Ours	0.101	0.130	0.112	0.127	0.121

Table 7 Action prediction accuracy (expressed in percent)

Object	Cup	Stone	Sponge	Spoon	Knife	Avg.
Vision	82.4	61.4	61.6	62.6	73.3	68.3
V + F	88.2	75.1	59.1	57.5	72.7	70.5

Comparsion of prediction using vision data only ("Vision") against using vision and force data ("V + F")

7 Conclusions and Future Work

In this paper, we proposed an approach to action interpretation, which treats the problem as a continuous updating of beliefs and predictions. The ideas were implemented for two tasks: the prediction of perceived action from visual input, and the prediction of force values on the hand. The methods were shown to run in real-time and demonstrated high accuracy performance. The action prediction was evaluated also against human performance and was shown to be nearly on

par. Additionally, new datasets of videos of dexterous actions and force measurements were created, which can be accessed from Fermüller (2016).

The methods presented here are only a first implementation of a concept that can be further developed along a number of directions. Here, we applied learning on 2D images only and, clearly, this way we also learn properties of the images that are not relevant to the task, such as the background textures. In order to become robust to these 'nuisances', 3D information, such as contours and depth features, could be considered in future work. While the current implementation only considers action labels, the same framework can be applied for other aspects of action understanding. For example, one can describe the different phases of actions and predict these sub-actions, since different actions share similar small movements. One can also describe the movements of other body parts, e.g., the arms and shoulders. Finally, the predicted forces may be used for learning how to perform actions on the robot. Future work will attempt to map the forces from the human hands onto other actuators, for example three-fingered hands or grippers.

Acknowledgements This work was funded by the support of the National Science Foundation under Grant SMA 1540917 and Grant CNS 1544797, by Samsung under the GRO Program (Nos. 20477, 355022), and by DARPA through U.S. Army Grant W911NF-14-1-0384.

References

Aloimonos, Y., & Fermüller, C. (2015). The cognitive dialogue: A new model for vision implementing common sense reasoning. *Image and Vision Computing*, *35*(12), 2891–2903.

Ansuini, C., Giosa, L., Turella, L., Altoé, G., & Castiello, U. (2008). An object for an action, the same object for other actions: Effects on hand shaping. *Experimental Brain Research*, *185*(1), 111–119.

Ansuini, C., Cavallo, A., Bertone, C., & Becchio, C. (2015). Intentions in the brain: The unveiling of Mister Hyde. *The Neuroscientist*, *21*(2), 126–135.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483.

Aviles, A. I., Marban, A., Sobrevilla, P., Fernandez, J., & Casals, A. (2014). A recurrent neural network approach for 3d vision-based force estimation. In *International conference on image processing theory, tools and applications (IPTA)*.

Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal Q2*.

Bullock, I. M., Feix, T., & Dollar, A. M. (2015). The Yale human grasping data set: Grasp, object, and task data in household and machine shop environments. *International Journal of Robotics Research. 34*(3), 251–255.

Cai, M., Kitani, K. M., & Sato, Y. (2015). A scalable approach for understanding the visual structures of hand grasps. In *IEEE international conference on robotics and automation (ICRA)*, IEEE (pp. 1360–1366).

Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *IEEE conference on computer vision and pattern recognition*, IEEE (Vol. 2, pp. 142–149).

Crajé, C., Lukos, J., Ansuini, C., Gordon, A., & Santello, M. (2011). The effects of task and content on digit placement on a bottle. *Exp Brain Research*, *212*(1), 119–124.

Cutkosky, M. R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, *5*(3), 269–279.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., et al. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *IEEE conference on computer vision and pattern recognition* (pp. 2625–2634).

Doyle, J., & Csete, M. (2011). Architecture, constraints, and behavior. *Proceedings of the National Academy of Sciences*, *108*(Sup. 3), 15624–15630.

Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., & Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, *108*(1), 52–73.

Fanello, S. R., Gori, I., Metta, G., & Odone, F. (2013). Keep it simple and sparse: Real-time action recognition. *The Journal of Machine Learning Research*, *14*(1), 2617–2640.

Fathi, A., Ren, X., & Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3281–3288).

Feix, T., Pawlik, R., Schmiedmayer, H., Romero, J., & Kragic, D. (2009). A comprehensive grasp taxonomy. In *Robotics, science and systems conference: Workshop on understanding the human hand for advancing robotic manipulation*.

Fermüller, C. (2016). *Prediction of manipulation actions*. http://www.cfar.umd.edu/~fer/action-prediction/.

Fouhey, D. F., & Zitnick, C. (2014). Predicting object dynamics in scenes. In *IEEE conference on computer vision and pattern recognition*.

Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). *Recurrent network models for kinematic tracking*. arXiv:1508.00271.

Gallesse, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, *2*(12), 493–501.

Gams, A., Do, M., Ude, A., Asfour, T., & Dillmann, R. (2010). On-line periodic movement and force-profile learning for adaptation to new surfaces. In *IEEE International conference on robotics research (ICRA)* (pp. 3192–3199).

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6645–6649).

Greminger, M., & Nelson, B. (2004). Vision-based force measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(3), 290–298.

Hoai, M., & De la Torre, F. (2014). Max-margin early event detectors. *International Journal of Computer Vision*, *107*(2), 191–202.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hoffman, J., Gupta, S., & Darrell, T. (2016) Learning with side information through modality hallucination. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Ijina, E., & Mohan, S. (2014). Human action recognition based on recognition of linear patterns in action bank features using convolutional neural networks. In *International conference on machine learning and applications*.

Jeannerod, M. (1984). The timing of natural prehension movements. *Journal of Motor Behavior*, *16*(3), 235–254.

Joo, J., Li, W., Steen, F. F., & Zhu, S. C. (2014). Visual persuasion: Inferring communicative intents of images. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 216–223).

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolu-

tional neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1725–1732).
Keskin, C., Kıraç, F., Kara, Y. E., & Akarun, L. (2013). Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision* (pp. 119–137). London: Springer.
Kitani, K. M., Ziebart, B. D., Bagnell, J. A., & Hebert, M. (2012). Activity forecasting. In *European conference on computer vision (ECCV)*.
Kober, J., Gienger, M., & Steil, J. (2000). Learning movement primitives for force interaction tasks. *IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 142–149.
Koppula, H., & Saxena, A. (2016). Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on pattern Analysis and Machine Intelligence*, *38*(1), 14–29.
Kormushev, P., Calinon, S., & Caldwell, D. G. (2011). Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, *25*(5), 581–603.
Lea, C., Reiter, A., Vidal, R., & Hager, G. D. (2016). Segmental spatiotemporal CNNS for fine-grained action segmentation. In *European conference of computer vision (ECCV)* (pp. 36–52).
Lenz, I., Lee, H., & Saxena, A. (2015) Deep learning for detecting robotic grasps. *The International Journal of Robotics Research. 34*(4–5), 705–724.
Li, Y., Fermüller, C., Aloimonos, Y., & Ji, H. (2010). Learning shift-invariant sparse representation of actions. In *IEEE conference on computer vision and pattern recognition*, San Francisco, CA, pp. 2630–2637.
Liu, J., Feng, F., Nakamura, Y. C., & Pollard, N. S. (2014). A taxonomy of everyday grasps in action. In *14th IEEE-RAS international conference on humanoid robots*, Humanoids.
Lv, F., & Nevatia, R.(2006). Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *European conference on computer vision (ECCV)* (pp. 359–372). Springer.
Ma, S., Sigal, L., & Sclaroff, S. (2016). Learning activity progression in LSTMS for activity detection and early detection. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
Mandary, C., Terlemez, O., Do, M., Vahrenkamp, N., & Asfour, T. (2015). The kit whole-body human motion database. In *International conferences on advanced robotics* (pp. 329–336).
Melax ,S., Keselman, L., & Orsten, S. (2013). Dynamics based 3d skeletal hand tracking. In *Proceedings of graphics interface 2013, Canadian information processing society* (pp. 63–70).
Moeslund, T., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2), 90–126.
Molchanov, P., Gupta, S., Kim. K., & Kautz, J. (2015). Hand gesture recognition with 3d convolutional neural networks. In *IEEE conference on computer vision and pattern recognition workshops* (pp. 1–7).
Ng, J. Y. H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga. R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *CVPR 2015*.
Ohn-Bar, E., & Trivedi, M. M. (2014). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, *15*(6), 2368–2377.
Oikonomidis, I., Kyriazis, N., & Argyros, A. A. (2011). Efficient model-based 3d tracking of hand articulations using kinect. In *British machine vision conference*.
Panteleris, P., Kyriazis, N., & Argyros, A. (2015). 3d tracking of human hands in interaction with unknown objects. In *British machine vision conference (BMVC)*, BMVA Press, pp. 123.
Pham, T. H., Kheddar, A., Qammaz, A., & Argyros, A. A. (2015). Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
Pieropan, A., Ek, C. H., & Kjellstrom, H. (2013). Functional object descriptors for human activity modeling. In *IEEE international conference on robotics and automation (ICRA)* (pp. 1282–1289).
Pirsiavash, H., Vondrick, C., & Torralba, A. (2014). *Inferring the why in images*. arXiv:1406.5472
Rizzolatti, G., Fogassi, L., & Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, *2*(9), 661–670.
Rogez, G., Khademi, M., Supančič, J. III, Montiel, J. M. M., & Ramanan, D. (2014). 3d hand pose detection in egocentric RGB-D images. In *Workshop at the European conference on computer vision* (pp. 356–371). Springer.
Rogez, G., Supancic, J. S. III., & Ramanan, D. (2015). Understanding everyday hands in action from RGB-D images. In *IEEE international conference on computer vision (ICCV)*.
Romero, J., Feix, T., Ek, C. H., Kjellstrom, H., & Kragic, D. (2013). A metric for comparing the anthropomorphic motion capability of artificial hands. *IEEE Transactions on Robotics*, *29*(6), 1342–1352.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.
Ryoo, M. S. (2011). Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE international conference on computer vision (ICCV)*.
Ryoo, M. S., & Matthies, L. (2013). First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2730–2737).
Ryoo, M., Fuchs, T. J., Xia, L., Aggarwal, J., & Matthies, L. (2015). Robot-centric activity prediction from first-person videos: What will they do to me? In *ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 295–302). ACM.
Saxena, A., Driemeyer, J., & Ng, A. Y. (2008). Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, *27*(2), 157–173.
Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *International conference on pattern recognition*.
Shi, Q., Cheng, L., Wang, L., & Smola, A. (2011). Human action segmentation and recognition using discriminative semi-Markov models. *International Journal of Computer Vision*, *93*(1), 22–32.
Shimoga, K. B. (1996). Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, *15*(3), 230–266.
Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., et al. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116–124.
Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv:1409.1556.
Starner, T., Weaver, J., & Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1371–1375.
Stein, S., & McKenna, S. J. (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on pervasive and ubiquitous computing* (pp. 729–738). ACM.
Supancic, J. S., Rogez, G., Yang, Y., Shotton, J., & Ramanan, D. (2015). Depth-based hand pose estimation: Data, methods, and challenges. In *IEEE international conference on computer vision* (pp. 1868–1876).
Takano, W., Ishikawa, J., & Nakamura, Y. (2015). Using a human action database to recognize actions in monocular image sequences: Recovering human whole body configurations. *Advanced Robotics*, *29*(12), 771–784.

Tiest, W. M. B., & Kappers, A. M. (2014). Physical aspects of softness perception. In M. D. Luca (Ed.), *Multisensory softness* (pp. 3–15). London: Springer.

Turaga, P., Chellappa, R., Subrahmanian, V., & Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1473–1488.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., & Saenko, K. (2014). *Translating videos to natural language using deep recurrent neural networks*. arXiv:1412.4729.

Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A., & Bengio, Y. (2015). *A recurrent neural network based alternative to convolutional networks*. arXiv:1505.00393.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Anticipating visual representations from unlabeled video. In *IEEE conference on computer vision and pattern recognition*.

Walker, J., Gupta, A., & Hebert, M. (2014). Patch to the future: Unsupervised visual prediction. In *IEEE conference on computer vision and pattern recognition* (pp. 3302–3309).

Wang, S. B., Quattoni, A., Morency, L. P., Demirdjian, D., & Darrell, T. (2006). Hidden conditional random fields for gesture recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 1521–1527.

Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2014). Learning actionlet ensemble for 3d human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(5), 914–927.

Xie, D., Todorovic, S., & Zhu, S. C. (2013). Inferring "dark matter" and "dark energy" from videos. In *IEEE international conference on computer vision (ICCV)* (pp. 2224–2231).

Yang, Y., Fermüller, C., Li, Y., & Aloimonos, Y. (2015). Grasp type revisited: A modern perspective on a classical feature for vision. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Zhu, Y., Zhao, Y., & Zhu, S. C. (2015). Understanding tools: Task-oriented object modeling, learning and recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 2855–2864).