This paper has been accepted for publication at the International Symposium on Robotics Research (ISRR), Hanoi, 2019.

Learning Collaborative Action Plans from YouTube Videos

Hejia Zhang, Po-Jen Lai, Sayan Paul,
Suraj Kothawade and Stefanos Nikolaidis

Department of Computer Science, University of Southern California Los Angeles 90089, USA, {hejiazha, pojenlai, sayanpau, kothawad, nikolaid}@usc.edu

Abstract. Videos from the World Wide Web provide a rich source of information that robots could use to acquire knowledge about manipulation tasks. Previous work has focused on generating action sequences from unconstrained videos for a single robot performing manipulation tasks by itself. However, robots operating in the same physical space with people need to not only perform actions autonomously, but also coordinate seamlessly with their human counterparts. This often requires representing and executing *collaborative* manipulation actions, such as handing over a tool or holding an object for the other agent. We present a system for knowledge acquisition of collaborative manipulation action plans that outputs commands to the robot in the form of visual sentence. We show the performance of the system in 12 unlabeled action clips taken from collaborative cooking videos on YouTube. We view this as the first step towards extracting collaborative manipulation action sequences from unconstrained, unlabeled online videos.

1 Introduction

We focus on the problem of learning manipulation actions for a robot, which operates as part of a human-robot team. We are particularly interested in the problem of learning actions from unstructured demonstrations by human teams. Leveraging the increasingly vast amount of content online, we envision a robot "watching" a video of humans performing a collaborative task, for instance preparing a meal together or assembling IKEA furniture, and then having the robot execute the same task alongside its human counterpart.

We therefore address the following research question:

> How can we learn collaborative manipulation plans from unconstrained videos of human teams?

This is challenging; online, unconstrained videos lack 3D information and suffer from poor lighting, occlusion and changing viewpoints. Additionally, our

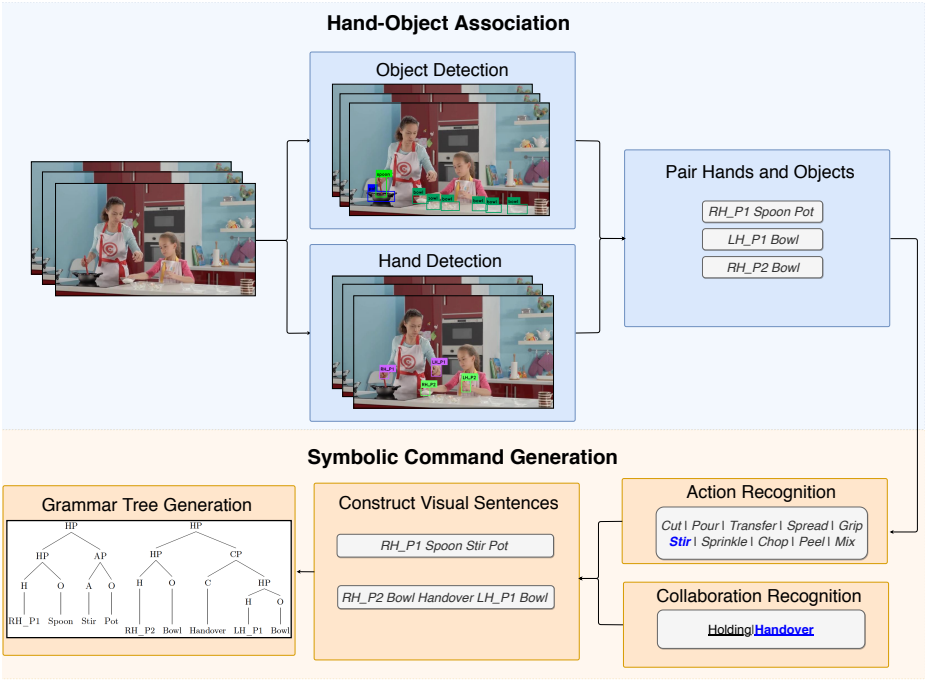


Fig. 1: System Architecture

task requires not only recognizing actions in the video, but also recognizing the factors that would enable each action to be executed by a collaborative robot, such as Sawyer: the tool used, and the object to be manipulated. For example, to cut a tomato, the robot would need to know that it needs to grasp a knife with one hand and apply the action to the tomato, grasped with the other hand. In collaborative manipulation actions, agents share the same objects and tools: the robot would need to know that it needs to hand over the knife, or that it needs to hold the pot so that the human teammate can stir it with the spoon.

In previous work [1], Yang et al. addressed the problem of learning action plans from unconstrained videos by proposing a system consisting of two levels: a lower level subsystem for extracting objects, hands and grasp types from videos, and a higher level for generating action commands using a manipulation grammar. This allowed robust generation of action commands in the cooking dataset YouCook [2]. The system focused on learning action plans for a single robot operating autonomously, using close-up videos of one human interacting with two objects.

In this paper we generalize this exciting work by implementing a collaborative manipulation action grammar which accounts for activities involving an arbitrary number of humans and objects, as well as physical interactions between two persons that manipulate the same object. This allows for processing a wider range of online content.

For visual processing of the videos, we leverage recent advances in object and human pose detection to extract perceptible elements. Specifically, we use YOLOv3 [3] for object detection and OpenPose [4, 5] for hand detection and association of hands with persons. Since hands are the main source of manipulation actions [6], we pair them with the detected objects using a distance threshold. Given the confidence on the labels of the manipulated objects, we use a language corpus [7] to predict the most likely actions. When two users manipulate the same object, we track whether the ownership of an object changes, to infer two types of collaborative actions: object holding and handover. Following a bottom-up approach, the system builds a visual sentence [1] and uses a probabilistic action grammar that accounts for collaborative actions to build a semantic, human-interpretable tree structure, that we can use as input command to the robot.

We demonstrate the pipeline on a YouTube playlist of unconstrained videos with two human teammates interacting in the same physical space on a cooking task. On unlabeled test clips of different collaborative scenarios, we show that our system can correctly generate a variety of visual sentences in 10 out of 12 clips. While creating a full action sequence from the whole video would require automatic clip segmentation and better object recognition, we find this to be an exciting first step for extracting meaningful semantic representations of collaborative action plans from unconstrained, unlabeled online videos.

2 Related Works

Learning from demonstration (LfD) provides a promising way for robots to learn manipulation tasks efficiently by mimicking expert policies [8]. LfD algorithms typically require human supervision, which can be labor intensive [9]. Researchers have developed datasets for robot learning tasks that allow large-scale data collection [10]. On the other hand, the rapidly growing online video content provides a rich source of knowledge that robots can acquire [1, 11].

There has been a lot of work on human activity recognition [12]. Recent work on deep learning approaches has enabled the generation of natural language [13], individual robot commands [14], and neural programs [15] using manually annotated datasets. Generalization is an important challenge in robot learning and to address this issue, Pastra et al. [16] discuss a Chomskyan grammar for understanding complex actions as a theoretical concept, inspired by Chomsky's suggestion that a minimalist generative grammar also exists for action understanding and execution [17]. Ryoo and Aggarwal propose a context-free grammar based scheme to recognize human activities, using a hierarchical structure with manually designed activity representations at the top level and human movements as building blocks at the bottom level [18, 19].

The work that is closest to ours is by Yang et al. [1]. The researchers have proposed a cognitive system which can directly learn manipulation actions from unconstrained YouTube videos, leveraging a set of context-free grammar rules for manipulation action understanding and deep neural networks for percep-

Fig. 2: Examples of object detection and hand detection result

tion. Their work focuses on inferring action plans from single-person videos. Summers-Stay et al. [20] provide an implementation of such a grammar, while Yang et al. [6] propose a set of context free grammar rules for manipulation action understanding. Leveraging the latest development in deep learning [3] and human body detection [5], we generalize to collaborative tasks for a robot interacting with a human teammate.

Learning collaboration actions has been also explored by Shu et al. [21], where a variety of social activities, represented as social affordances, are learned in the form of spatiotemporal AND-OR Graphs. Their approach focuses on low-level joint trajectories, using RGB-D data of human demonstrations in a lab setting, and it generalizes previous work on learning object affordances [22,23]. Similarly, Amor et al. [24] propose to learn interaction motion primitives from joint activities of two human actors, recorded with a motion capture system. Instead, our focus is on learning symbolic command sequences using unconstrained videos.

3 Our Approach

Our system consists of two main components: a Deep Neural Network (DNN) based visual processing subsystem and a collaborative manipulation action grammar based parser. The visual processing subsystem manages the detection and association of hands with objects (Fig. 1). The second component processes the detected objects and hands to infer the actions performed by each person, as well as the collaborative actions performed by two persons together. Using this information, it constructs visual sentences and generates symbolic commands in the form of grammar trees.

3.1 Hand-Object Association

The design of our system is motivated by two observations [6]: (1) Most manipulation actions follow a hierarchy where objects are either tools operating on other objects or they are being operated by other objects; (2) Hands are the main driving force in manipulation actions. Based on these observations, we start by detecting hands and objects, and infer which objects are manipulated by each hand.

Algorithm 1 Pair Hands and Objects

```
1: procedure PAIR(F)
2:     HO ← ∅
3:     %%%% Pair objects and hands based on distance and given thresholds
4:     for each frame f in F do
5:         ho ← ∅
6:         for each hand h detected in f do
7:             if Found closest object o (within threshold) then
8:                 ho.Append(h, o)
9:         HO.Append(ho)
10:    %%%% Filtering pairing results
11:    for each hand h do
12:        for each frame f in F do
13:            if h is not paired with object for consecutive K1 frames
                   starting from f then
14:                ho ← hand-object pairs from HO for frame f
15:                ho.Remove(h's pair)
16:    %%%% Pair tool objects gasped by hands with manipulated objects
17:    for each frame f in F do
18:        hoo ← ∅
19:        for each hand h detected in f do
20:            if h doesn't pair with any object on f then
21:                continue
22:            ho ← hand-object pairs from HO at frame f
23:            tool_object ← ho.get_paired_object(h)
24:            if Found closest object o manipulated by
                   tool_object (within threshold) then
25:                hoo.Append(h, tool_object, o)
26:        HOO.Append(hoo)
27:    %%%% Filtering pairing results
28:    for each hand h do
29:        for each frame f in F do
30:            if h is not paired with two objects for consecutive K2
                   frames starting from f then
31:                hoo ← hand-tool-object tuple from HOO at frame f
32:                hoo.Remove(h's tuple)
33:    %%%% Normalize object detection results
34:    for each hand h do
35:        HO.Normalize_Object_Detection_Results(h)
36:        HOO.Normalize_Object_Detection_Results(h)
37:    return HO, HOO
```

DNN based Visual Perception. Our DNN based visual perception subsystem has two visual detection modules: one for detecting human hands and the other for detecting objects. In both modules we use state-of-the-art detection frameworks.

The hand detection component takes as input a video clip and it outputs bounding boxes, labels (left hand or right hand), and confidences for each hand in each frame. In unconstrained videos, viewpoint changes and occlusions are frequent. We use the OpenPose human body keypoint detection framework [4,5], which accurately associates human hands with individuals.

The object detection component outputs bounding boxes and probabilities for each class of the objects in our testing set. We use YOLOv3 [3] for extracting the location as well as a confidence score for each object class for the individual bounding boxes. YOLOv3 is built upon previous YOLO architectures [25,26] and belongs to the class of systems known as single shot detectors. It uses feature pyramids to detect small objects in images. We use a pre-trained model trained on the COCO dataset [27] for transfer learning and fine-tune it with a custom object dataset.

Hand and Object Pairing. Based on the object and hand detection results and our assumption of manipulation hierarchy, we perform two rounds of hand and object pairing (Alg. 1).

In the first round, we find objects that are grasped directly by hands based on the distance between the bounding box of detected objects and the bounding box of detected hands. To each hand we associate the nearest object, if the distance between the two is below a predefined threshold (Alg. 1, lines 4–9).

In the second round, we check whether the objects paired with hands in the first round are used as tools to manipulate other objects. For instance, a knife could be used as a tool operating on a tomato. Similarly to the hand-object association, we correspond an object with a second object using a distance metric (lines 17–26).

To improve the robustness of the system, we filter the results from each pairing round, i.e. we only keep the hand and object pairs that are retained for consecutive K_i (K_1 for the first round and K_2 for the second round) frames (lines 11–15, lines 28–32).

Finally, along the temporal dimension, we normalize the belief distributions of the labels of the objects that are associated with other objects or hands (lines 34–36).

3.2 Grammar Tree Generation

Once we have paired hands and objects for each frame in video clips of interest, we construct visual sentences (Alg. 2), which will then generate grammar trees based on our collaborative manipulation action context-free grammar rules.

Corpus Guided Action Recognition. Because of the large variations of manipulation actions, especially in collaboration settings, visual activity recognition is a challenging problem. Instead, we follow the approach of Yang et al. [1] using a trained language model and the normalized object class predictions from Alg. 1. The corpus we use for our language model is the one billion word corpus [28], from which we consider only the words present in a pre-specified object and action set. For each possible trigram, we extract $P(Action \mid Object1, Object2)$ and compute the action probabilities as follows:

$$P(Action) = \sum_{Object1,Object2} P(Action \mid Object1, Object2) \times P_{Object1} \times P_{Object2} \quad (1)$$

Collaboration Recognition. We detect a collaboration when: 1) we find two persons grasping the same object, or 2) the object grasped by one person is used as a tool to manipulate an object grasped by the other person. This is illustrated in Alg. 2, lines 1–5. Contrary to single person atomic actions, which we can infer purely from spatial information, inference of different collaboration actions requires reasoning also in the temporal domain. We explore two types of such actions: "Handover" and "Holding".

We detect a "Handover" when two persons are grasping the same object, and the ownership of the object changes over time (lines 6–12). We detect "Holding" when the ownership does not change, or when a person is using one object to manipulate an object grasped by the other person.

Similarly to Alg. 1, we recognize when the interactions persist for a prespecified number of frames, to reduce false positives (Alg. 2, line 19).

Algorithm 2 Construct Visual Sentences

```
1: procedure InteractingObject(o_1, o_2)
2:     if (o_1 = o_2) or (o_1 manipulates o_2) or (o_2 manipulates o_1) then
3:         return True
4:     else
5:         return False
6: procedure IsHandover(F, o)
7:     start_owner ← the first person owns o
8:     end_owner ← the last person owns o
9:     if start_owner ≠ end_owner then
10:        return True
11:    else
12:        return False
13: procedure Construct(F, HO, HOO)
14:    sentences ← []
15:    %%%% Construct visual sentences invloving collaboration
16:    for each hand h_1 of first person do
17:        for each hand h_2 of the second person do
18:            for each frame f in F do
19:                if (h_1 and h_2 are paired with objects o_1 and o_2 in HO ) and
                       (InteractingObject(o_1, o_2)) for consecutive q_1 frames then
20:                    if o_1 ≠ o_2 then
21:                        interaction_type ← "Holding"
22:                        holder ← the person whose object is being manipulated
23:                        user ← the other person
```

action ← get action from corpus
sentences.Append(*holder*, *holder*'s *object*, *interaction_type*, *user*, *user*'s *object*, *action*, *holder*'s *object*)
break
else if IsHandover(F, o_1) **then**
interaction_type ← "*Handover*"
sentences.Append(*start_owner*, *start_owner*'s *object*, *interaction_type*, *end_owner*, *end_owner*'s *object*)
break
else
continue
%%%% Construct visual sentences invloving both hands of one person
for each person P **do**
h_1 ← P's one hand
h_2 ← P's other hand
for each frame f in F **do**
if (h_1 and h_2 are paired with objects o_1 and o_2 in HO) and ((o_1 manipulates o_2) or (o_2 manipulates o_1)) for consecutive q_2 frames **then**
manipulated_hand ← the hand of the object being manipulated
manipulator_hand ← the other hand
action ← get action from corpus
sentences.Append(*manipulator_hand*, *manipulator_hand*'s *object*, *action*, *manipulated_hand*, *manipulated_hand*'s *object*)
break
%%%% Construct visual sentences invloving one hand of one person
for each person P **do**
for each hand h of P **do**
if h is paired with two objects in HOO **then**
object_tool ← the object paired directly with h
object ← the other object
action ← get action from corpus
sentences.Append(h, *object_tool*, *action*, *object*)
return *sentences*

Visual Sentence Generation Once we have recognized actions and collaboration actions, the system constructs a visual sentence based on the number of objects assigned to each hand and the computed probabilities of objects and actions (Alg. 2). The visual sentences capture different cases of actions and interactions, where two persons operate on the same object (items 1, 2 below), or one person uses an object as a tool on another object (items 3, 4):

1. (LeftHandPerson1, Object1, CollaborativeAction, LeftHandPerson2, Object2, Action, RightHandPerson2, Object1): Persons 1 and 2 collaborate by operating on the same object. Person 1 grasps Object1, while person 2 uses Object

$$
\begin{array}{lcll}
HP & \rightarrow & H\ O \mid HP\ AP \mid HP\ CP & 0.33 \quad (1) \\
AP & \rightarrow & A\ O \mid A\ HP & 0.5 \quad (2) \\
CP & \rightarrow & C\ HP & 1.0 \quad (3) \\
H & \rightarrow & \text{“}LH\_P1\text{”} \mid \text{“}RH\_P1\text{”} \mid \text{“}LH\_P2\text{”} \mid \text{“}RH\_P2\text{”} & 0.25 \quad (4) \\
C & \rightarrow & Collaboration & P_{Collaboration} \quad (5) \\
O & \rightarrow & Object & P_{Object} \quad (6) \\
A & \rightarrow & Action & P_{Action} \quad (7)
\end{array}
$$

Fig. 3: A Probabilistic Collaborative Manipulation Action Context-Free Grammar

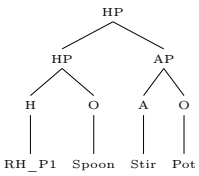
2 as a tool to perform an action on Object1 (Alg. 2, lines 20–26), which is also grasped by Person 1.

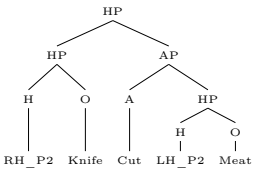
2. (LeftHandPerson1, Object1, CollaborativeAction, LeftHandPerson2, Object1): Persons 1 and 2 grasp the same object (Object 1), for instance during a handover (Alg. 2, lines 27–30).
3. (LeftHandPerson1, Object1, Action, RightHandPerson1, Object2): Person 1 uses Object1 as a tool to perform an action on Object2 grasped by their other hand, e.g., using a knife to cut a tomato (Alg. 2, lines 34–43).
4. (LeftHandPerson1, Object1, Action1, Object2): same as above, but Object 2 is not grasped, e.g., grasping a spoon to stir a pot (Alg. 2, lines 45–51).

The left and right hands in the visual sentences above can be interchanged.

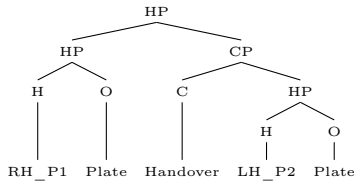
Collaborative Manipulation Action Grammar and Parsing. At the highest level of the system, we generate a symbolic command sequence using an extension of the probabilistic context-free grammar proposed by Yang et al. [1]. To account for collaboration between agents we extend the grammar by adding a collaboration terminal (C) and a collaboration phrase (CP) with a production rule: $HP \rightarrow HP\ CP$ where HP represents a hand phrase. The grammar assumes that hands (H) are the driving force of both single manipulation actions (A) and collaborative actions (C). A hand phrase (HP) contains an object (O), an action phrase (AP), or a collaborative action phrase (CP). The latter applies a collaborative action to a hand phrase.

Similar to Yang et al. [1], we treat each sub-rule in rules (1), (2) and (3) equally, assigning them equal probabilities. With regard to rule (4), since we consider activities involving two humans, we have four different hands, i.e. “LH_P1”, “RH_P1”, “LH_P2” and “RH_P2” with equal probability. For the terminal rules (6–7), we assign the normalized belief distributions obtained from the visual process subsystem. For the terminal rule (5), collaboration can be “Handover” or “Holding” with probability 1, based on the deterministic output of Alg. 2. We use a Viterbi parser to parse the constructed visual sentences and output the most likely parse tree of the specific manipulation action. The robot can then execute the plan by reversely parsing the tree.

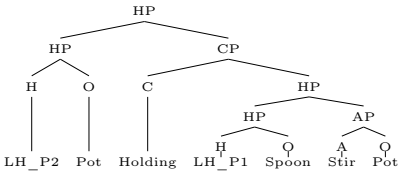





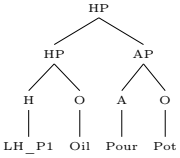
(a) The woman is stirring the pot while the girl is cutting the meat.

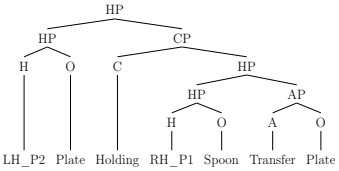
(b) The girl is handing over the plate to the woman.

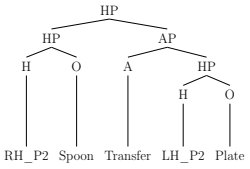



(c) The woman is holding the pot for the girl to stir.

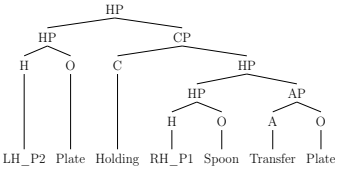



(d) The woman is pouring oil to the pot.

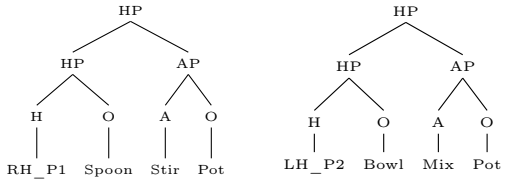


(e) The woman is holding the plate, while herself and the girl are transferring food.

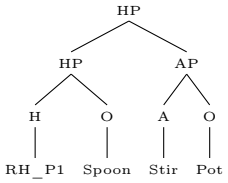



(f) The girl is holding the plate for the woman to transfer food.

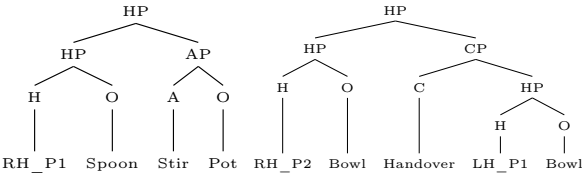
Fig. 4: Example frames of each test clip and generated parse trees. The captions depict the ground-truth descriptions of each test clip (cont'd on the next page).

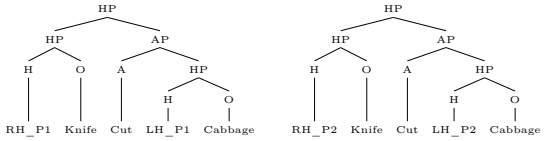
(g) The woman is stirring the pot while the girl is mixing the food to the pot.

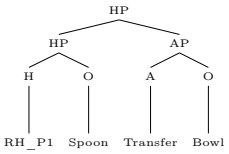
(h) The woman is stirring the pot.

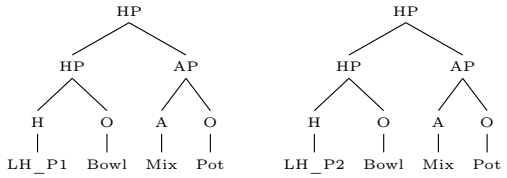
(i) The girl is handing over a bowl to the woman who is stirring the pot.

(j) The man and the boy are cutting cabbage.

(k) Failure case: the woman is spreading meat on the lasagne.

(l) Failure case: The woman is not touching the bowl.

Fig. 4: Example frames of each test clip and generated parse trees. The captions depict the ground-truth descriptions of each test clip.

4 Experiments

We focus on a collaborative cooking scenario, using YouTube videos of two persons cooking together. We are interested in whether the automatically generated trees from the unlabeled video clips capture the single and collaborative actions performed in the videos.

Setting. We tested our framework on the publicly available mini chef Youtube playlist.[1] The dataset has a rich variety of different scenarios: both persons acting independently, one person acts while the other is watching, a person hands over an object to the other person, one person holds an object while the other person performs an action on that object, one person holds the object while they both perform an action on that object. All these scenarios are captured in the visual sentence representations generated by Alg. 2.

We note that while this dataset includes videos with only two persons, our framework can be used for an arbitrary number of humans performing tasks independently and collaborating in pairs, by generating a tree for each individual action or each interaction.

We selected 12 test clips out of 7 videos of the playlist to cover all scenarios, each of approximately 100 frames (∼3 seconds). We fine-tuned the pre-trained YOLOv3 model with manually labeled object classes from a training set of 15334 annotated frames that were not included in the test clips. To mitigate the class-imbalance problem we utilized the cut-paste method [29] for training.

Fig. 5 shows example frame sequences, with the overlayed bounding boxes of the hand-object associations derived with Alg. 1. Similarly to Yang et al. [1], our action set was: (Cut, Pour, Transfer, Spread, Grip, Stir, Sprinkle, Chop, Peel, Mix).

Results. Fig. 4 shows the generated trees for each test-clip. The algorithm generated correct trees in 10 out of the 12 test-cases. Successful test cases and their action/collaboration types are summarized in Table 1.

For instance, in Fig. 4(a), we can first see that the woman is grasping a spoon used on a pot while the girl is grasping a knife with the right hand and meat with the left hand. For the woman, the action derived from the language corpus for the detected objects "spoon" and "pot" is "stir". For the girl, the derived action is "cut". The generated tree describes that the woman is stirring something in the pot with a spoon while the girl is using a knife to cut meat. Because they are working independently, there are two independent trees describing their actions.

Fig. 4(e) presents a more complicated case, where the woman is holding a plate for the girl to transfer food. At the same time, the woman is also grasping the spoon perform the transfer. For this scenario, our system generates two trees. The first tree represents that the woman is holding a plate for the girl to transfer food, while the second tree describes that the woman is also using the spoon to transfer food.

Our system fails in two scenarios, which are shown in Fig. 4(k) and Fig. 4(l). In Fig. 4(k), the woman is not transferring the meat. Instead, she is spreading

[1] https://www.youtube.com/playlist?list=PL1204B2E3981AF56E

Action/Collaboration Type	Result
Two persons acting independently	4(a), 4(g), 4(j)
Two persons collaborating	4(b)
One person doing an action while another person is collaborating	4(c), 4(f), 4(i)
Only one person doing an action	4(d), 4(h)
Two persons doing an action together, while one person is collaborating	4(e)

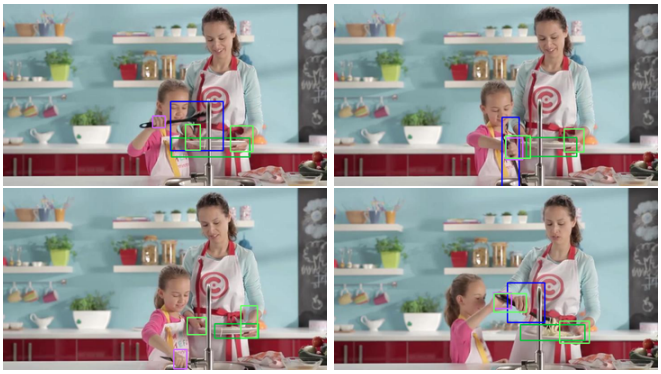
Table 1: The action/collaboration types shown in Fig. 4

meat on the lasagne. The reason this case fails is that the action she performs does not match the action derived from the corpus. In Fig. 4(l), the woman is not grasping the bowl, even though it appears so in several frames because of the overlap. However, our system mistakenly associates the bowl with the woman's hand, and generates a non-existent action.

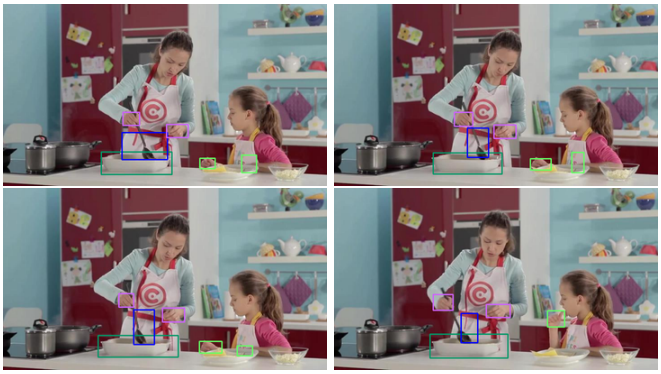
(a) The woman is stirring pot while the girl is cutting meat.

(b) The girl hands over the plate to the woman.

Fig. 5: Example frame sequences of test clips (cont'd on the next page).



(c) The woman is holding the plate for the girl to transfer food.



(d) The woman is spreading the meat on the lasagne.

Fig. 5: Example frame sequences of test clips.

5 Conclusion

We proposed a framework for robots to learn collaborative manipulation action plans from unconstrained videos. By using state-of-the-art object and hand detection algorithms, the system can make detections even in cases where visibility is low. The system exploits the spatial and temporal structure of collaborative actions to recognize object holdings and handovers and to construct visual sentences. Finally, the system parses the visual sentences to generate grammar trees that represent collaborative action plans.

We conducted experiments on a collaborative cooking dataset which consists of unconstrained demonstration videos of two persons cooking together. We selected 12 clips from this dataset to test our system, choosing scenarios of different individual and collaborative behaviors. The experiments showed that our system generates correct grammar tree descriptions for most of these scenarios.

Our system is limited in many ways. While the YouTube playlist is unconstrained, the cooking videos are meant to be instructive. This results in many actions being explicit and interpretable. On one hand, we expect that the language corpus-based action prediction would be robust to implicit actions as well,

since it relies only on robust object detections. On the other hand, such predictions are limited to "commonsense" instances and will fail in infrequent actions, such as cutting food with a spoon instead of a knife. Ultimately, we wish to integrate the generated grammar tree descriptions into a complete plan that allows a robot to execute the task with a human teammate, and we are excited about the challenges and opportunities that arise from transferring the learned actions into a physical workspace.

Overall, we find this is an exciting first step towards learning collaborative action plans from unconstrained, unlabeled online videos. On top of learning other collaborative actions, we are also interested in learning the high-level goal that drives the actions. For example, if one person is grasping a spoon to stir tomatoes in a pot, that is because this is required to make a tomato sauce. If robots can understand the motivation for each action, they can then actively help another agent to achieve the goal instead of blindly following the steps portrayed in the videos. We believe this is an important topic to explore for cognitive collaborative robots.

References

1. Yang, Y., Li, Y., Fermüller, C., Aloimonos, Y.: Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In: AAAI. pp. 3686–3693 (2015)
2. Das, P., Xu, C., Doell, R.F., Corso, J.J.: A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2634–2641 (2013)
3. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
4. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: CVPR (2017)
5. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In: arXiv preprint arXiv:1812.08008 (2018)
6. Yang, Y., Guha, A., Fermuller, C., Aloimonos, Y.: A cognitive system for understanding human manipulation actions. Advances in Cognitive Sysytems 3, 67–86 (2014)
7. Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., Robinson, T.: One billion word benchmark for measuring progress in statistical language modeling. In: INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association. pp. 2635–2639 (2014)
8. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and autonomous systems 57(5), 469–483 (2009)
9. Zhang, H., Heiden, E., Nikolaidis, S., Lim, J.J., Sukhatme, G.S.: Auto-conditioned recurrent mixture density networks for learning generalizable robot skills. CoRR abs/1810.00146 (2019)
10. Mandlekar, A., Zhu, Y., Garg, A., Booher, J., Spero, M., Tung, A., Gao, J., Emmons, J., Gupta, A., Orbay, E., Savarese, S., Fei-Fei, L.: ROBOTURK: A crowdsourcing platform for robotic skill learning through imitation. In: 2nd Annual Conference on Robot Learning, CoRL 2018. pp. 879–893 (2018)

11. Malmaud, J., Huang, J., Rathod, V., Johnston, N., Rabinovich, A., Murphy, K.: What's cookin'? interpreting cooking videos using text, speech and vision. CoRR abs/1503.01558 (2015), http://arxiv.org/abs/1503.01558
12. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video technology 18(11), 1473 (2008)
13. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R.J., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. In: NAACL HLT 2015. pp. 1494–1504 (2015)
14. Nguyen, A., Kanoulas, D., Muratore, L., Caldwell, D.G., Tsagarakis, N.G.: Translating videos to commands for robotic manipulation with deep recurrent neural networks. In: 2018 ICRA. IEEE (2018)
15. Sun, S.H., Noh, H., Somasundaram, S., Lim, J.: Neural program synthesis from diverse demonstration videos. In: ICML. pp. 4797–4806 (2018)
16. Pastra, K., Aloimonos, Y.: The minimalist grammar of action. Philosophical Transactions of the Royal Society B: Biological Sciences 367(1585), 103–117 (2012)
17. Chomsky, N.: Lectures on government and binding: The Pisa lectures. No. 9, Walter de Gruyter (1993)
18. Ryoo, M.S., Aggarwal, J.K.: Semantic representation and recognition of continued and recursive human activities. International journal of computer vision 82(1), 1–24 (2009)
19. Ryoo, M., Aggarwal, J.: Stochastic representation and recognition of high-level group activities. International journal of computer Vision 93(2), 183–200 (2011)
20. Summers-Stay, D., Teo, C.L., Yang, Y., Fermüller, C., Aloimonos, Y.: Using a minimal action grammar for activity understanding in the real world. In: IROS. pp. 4104–4111. IEEE (2012)
21. Shu, T., Gao, X., Ryoo, M.S., Zhu, S.: Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions. In: 2017 IEEE International Conference on Robotics and Automation, ICRA. pp. 1669–1676 (2017)
22. Koppula, H., Saxena, A.: Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In: ICML. pp. 792–800 (2013)
23. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. IJRR 32(8), 951–970 (2013)
24. Amor, H.B., Neumann, G., Kamthe, S., Kroemer, O., Peters, J.: Interaction primitives for human-robot cooperation tasks. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 2831–2837. IEEE (2014)
25. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2016)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
28. Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., Robinson, T.: One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005 (2013)
29. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1301–1310 (2017)