

---

# Robot Learning Manipulation Action Plans by “Watching” Unconstrained Videos

---

**Yezhou Yang, Cornelia Fermüller, Yiannis Aloimonos**  
University of Maryland  
{zyyang,fer,yiannis}@umiacs.umd.edu

**Yi Li**  
NICTA, Australia  
yi.li@nicta.com.au

## Abstract

In order to advance action generation and creation in robots beyond simple learned schemas we need computational tools that allow us to automatically interpret and represent human actions. This paper presents a system that learns manipulation action plans by processing unconstrained videos from the World Wide Web. Its goal is to robustly generate the sequence of atomic actions of seen longer actions in video in order to acquire knowledge for robots. The lower level of the system consists of two convolutional neural network (CNN) based recognition modules, one for classifying the hand grasp type and the other for object recognition. The higher level is a probabilistic manipulation action grammar based parsing module that aims at generating visual sentences for robot manipulation. Experiments conducted on a publicly available unconstrained video dataset show that the system is able to learn manipulation actions by “watching” unconstrained videos with high accuracy.

## 1 Introduction

The ability to learn actions from human demonstrations is one of the major challenges for the development of intelligent systems. Particularly, manipulation actions are very challenging, as there is large variation in the way they can be performed and there are many occlusions.

Our ultimate goal is to build a self-learning robot that is able to enrich its knowledge about fine grained manipulation actions by “watching” demo videos. In this work we explicitly model actions that involve different kinds of grasping, and aim at generating a sequence of atomic commands by processing unconstrained videos from the World Wide Web (WWW).

The robotics community has been studying perception and control problems of grasping for decades [12]. Recently, several learning based systems were reported that infer contact points or how to grasp an object from its appearance [11, 8]. However, the desired grasping type could be different for the same target object, when used for different action goals. Traditionally, data about the grasp has been acquired using motion capture gloves or hand trackers, such as the model-based tracker of [10]. The acquisition of grasp information from video (without 3D information) is still considered very difficult because of the large variation in appearance and the occlusions of the hand from objects during manipulation.

Our premise is that actions of manipulation are represented at multiple levels of abstraction. At lower levels the symbolic quantities are grounded in perception, and at the high level a grammatical structure represents symbolic information (objects, grasping types, actions). With the recent development of deep neural network approaches, our system integrates a CNN based object recognition and a CNN based grasping type recognition module. The latter recognizes the subject’s grasping type directly from image patches.

The grasp type is an essential component in the characterization of manipulation actions. Just from the viewpoint of processing videos, the grasp contains information about the action itself, and it can be used for prediction or as a feature for recognition. It also contains information about the beginning and end of action segments, thus it can be used to segment videos in time. If we are to perform the action with a robot, knowledge about how to grasp the object is necessary so the robot can arrange its effectors. For example, consider a humanoid with one parallel gripper and one vacuum gripper. When a power grasp is desired, the robot should select the vacuum gripper for a stable grasp, but when a precision grasp is desired, the parallel gripper is a better choice. Thus, knowing the grasping type provides information for the robot to plan the configuration of its effectors, or even the type of effector to use.

In order to perform a manipulation action, the robot also needs to learn what tool to grasp and on what object to perform the action. Our system applies CNN based recognition modules to recognize the objects and tools in the video. Then, given the beliefs of the tool and object (from the output of the recognition), our system predicts the most likely action using language, by mining a large corpus using a technique similar to [14]. Putting everything together, the output from the lower level visual perception system is in the form of (LeftHand GraspType1 Object1 Action RightHand GraspType2 Object2). We will refer to this septet of quantities as *visual sentence*.

At the higher level of representation, we generate a symbolic command sequence. [13] proposed a context-free grammar and related operations to parse manipulation actions. However, their system only processed RGBD data from a controlled lab environment. Furthermore, they did not consider the grasping type in the grammar. This work extends [13] by modeling manipulation actions using a probabilistic variant of the context free grammar, and explicitly modeling the grasping type.

Using as input the belief distributions from the CNN based visual perception system, a Viterbi probabilistic parser is used to represent actions in form of a hierarchical and recursive tree structure. This structure innately encodes the order of atomic actions in a sequence, and forms the basic unit of our knowledge representation. By reverse parsing it, our system is able to generate a sequence of atomic commands in predicate form, i.e. as  $Action(Subject, Patient)$  plus the temporal information necessary to guide the robot [1].

Our contributions are twofold. (1) A convolutional neural network (CNN) based method has been adopted to achieve state-of-the-art performance in grasping type classification and object recognition on unconstrained video data; (2) a system for learning information about human manipulation action has been developed that links lower level visual perception and higher level semantic structures through a probabilistic manipulation action grammar.

## 2 Our Approach

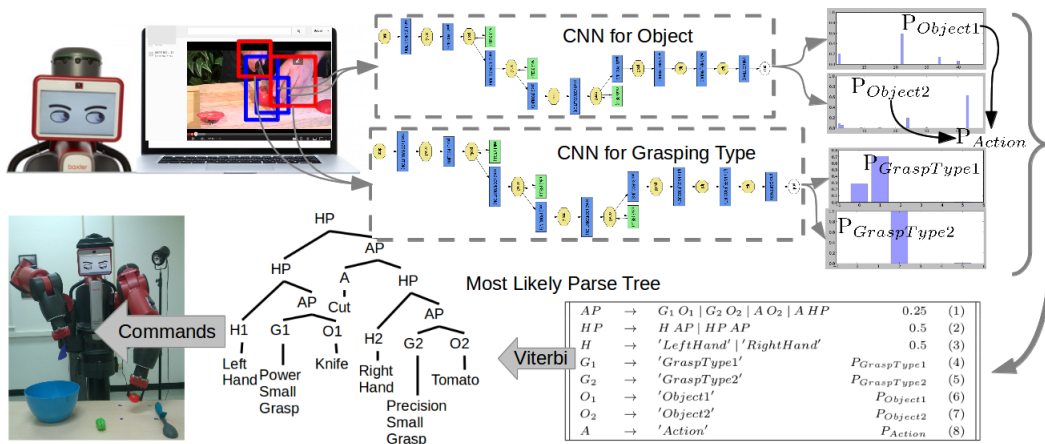


Figure 1: The integrated system reported in this work.

We developed a system to learn manipulation actions from unconstrained videos. The system takes advantage of: (1) the robustness from CNN based visual processing; (2) the generality of an action grammar based parser. Figure 1 shows our integrated approach.

In this work, we use a categorization into six grasping types. First we distinguish, according to the most commonly used classification (based on functionality) into power and precision grasps [6]. Power grasping is used when the object needs to be held firmly in order to apply force, such as “grasping a knife to cut”; precision grasping is used in order to do fine grain actions that require accuracy, such as “pinch a needle”. We then further distinguish among the power grasps, whether they are spherical, or otherwise (usually cylindrical), and we distinguish the latter according to the grasping diameter, into large diameter and small diameter ones. Similarly, we distinguish the precision grasps into large and small diameter ones. Additionally, we also consider a Rest position (no grasping performed).

We used a seven layer CNN (including the input layer and two perception layers for regression output). The first convolution layer has 32 filters of size  $5 \times 5$ , the second convolution layer has 32 filters of size  $5 \times 5$ , and the third convolution layer has 64 filters of size  $5 \times 5$ , respectively. The first perception layer has 64 regression outputs and the final perception layer has 6 regression outputs. Our system considers 6 grasping type classes.

Similar to the grasping type recognition module, we also used a seven layer CNN. The network structure is the same as before, except that the final perception layer has 48 regression outputs. Our system considers 48 object classes, and we denote this candidate object list as  $O$  in the rest of the paper. Given the beliefs of the tool and object (from the output of the recognition), our system predicts the most likely action using language, by mining a large corpus [14].

$AP$	$\rightarrow$	$G_1 O_1   G_2 O_2   A O_2   A HP$	0.25	(1)
$HP$	$\rightarrow$	$H AP   HP AP$	0.5	(2)
$H$	$\rightarrow$	$'LeftHand'   'RightHand'$	0.5	(3)
$G_1$	$\rightarrow$	$'GraspType1'$	$P_{GraspType1}$	(4)
$G_2$	$\rightarrow$	$'GraspType2'$	$P_{GraspType2}$	(5)
$O_1$	$\rightarrow$	$'Object1'$	$P_{Object1}$	(6)
$O_2$	$\rightarrow$	$'Object2'$	$P_{Object2}$	(7)
$A$	$\rightarrow$	$'Action'$	$P_{Action}$	(8)

Table 1: A Probabilistic Extension of Manipulation Action Context-Free Grammar.

At the higher level of representation, we generate a symbolic command sequence. This work extends [13] by modeling manipulations actions using a probabilistic variant of the context free grammar (Table 1), and explicitly modeling the grasping type. We use a bottom-up variation of the probabilistic context-free grammar parser that uses dynamic programming (best-known as Viterbi parser [4]) to find the most likely parse for an input visual sentence. The Viterbi parser parses the visual sentence by filling in the most likely constituent table, and the parser uses the grammar introduced in Table 1. For each testing video, our system outputs the most likely parse tree of the specific manipulation action. By reversely parsing the tree structure, the robot could derive an action plan for execution. Figure 3 shows sample output trees, and Table 2 shows the final control commands generated by reverse parsing.

### 3 Experiments

#### 3.1 Dataset and experimental settings

Cooking is an activity, requiring a variety of manipulation actions, that future service robots most likely need to learn. We conducted our experiments on a publicly available cooking video dataset collected from the WWW and fully labeled, called the Youtube cooking dataset (YouCook) [5]. The data was prepared from 88 open-source Youtube cooking videos with unconstrained third-person view. Frame-by-frame object annotations are provided for 49 out of the 88 videos. These features make it a good empirical testing bed for our hypotheses.

We conducted our experiments using the following protocols: (1) 12 video clips, which contain one typical kitchen action each, are reserved for testing; (2) all other video frames are used for training; (3) we randomly reserve 10% of the training data as validation set for training the CNNs.

For training the grasping type, we extended the dataset by annotating image patches containing hands in the training videos. The image patches were converted to gray-scale and then resized to  $32 \times 32$  pixels. The training set contains 1525 image patches and was labeled with the six grasping types. We used a GPU based CNN implementation [7] to train the neural network, following the structures described above. For training the object recognition CNN, we first extracted annotated image patches from the labeled training videos, and then resized them to  $32 \times 32 \times 3$ . We used the same GPU based CNN implementation to train the neural network, following the structures described above.

For localizing hands on the testing data, we first applied the hand detector from [9] and picked the top two hand patch proposals (left hand and right hand, if present). For objects, we trained general object detectors from labeled training data using techniques from [3]. Furthermore we associated candidate object patches with the left or right hand, respectively depending on which had the smaller Euclidean distance.

### 3.2 Grasping Type and Object Recognition

On the reserved 10% validation data, the grasping type recognition module achieved an average precision of 77% and an average recall of 76%. On the reserved 10% validation data, the object recognition module achieved an average precision of 93% and an average recall of 93%. Figure 2 shows the confusion matrices for grasping type and object recognition, respectively. From the figure we can see the robustness of the recognition.

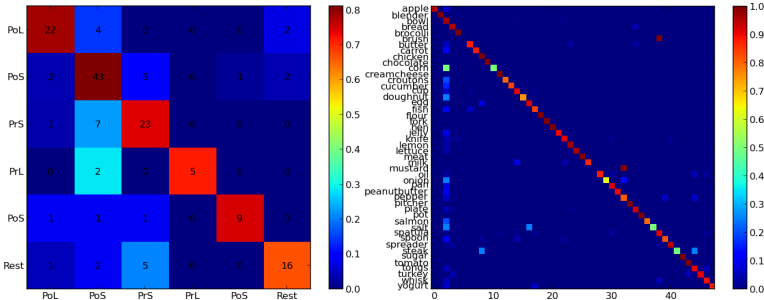


Figure 2: Confusion matrices (grasping type and object).

The performance of the object and grasping type recognition modules is also reflected in the commands that our system generated from the testing videos. We observed an overall recognition accuracy of 79% on objects, of 91% on grasping types and of 83% on predicted actions (see Table 2). It is worth mentioning that in the generated commands the performance in the recognition of object drops, because some of the objects in the testing sequences do not have training data, such as “Tofu”. The performance in the classification of grasping type goes up, because we sum up the grasping types belief distributions over the frames, which helps to smooth out wrong labels.

### 3.3 Visual Sentence Parsing and Commands Generation for Robots

Following the probabilistic action grammar from Table 1, we built upon the implementation of the Viterbi parser from the Natural Language Processing Kit [2] to generate the single most likely parse tree from the probabilistic visual sentence input. Figure 3 shows the sample visual processing outputs and final parse trees obtained using our integrated system. Table 2 lists the commands generated by our system on the reserved 12 testing videos, shown together with the ground truth commands. The overall percentage of correct commands is 68%. Note, that we considered a command predicate wrong, if any of the object, grasping type or action was recognized incorrectly.

## 4 Conclusion

In this paper we presented an approach to learn manipulation action plans from unconstrained videos for cognitive robots. Two convolutional neural network based recognition modules (for grasping

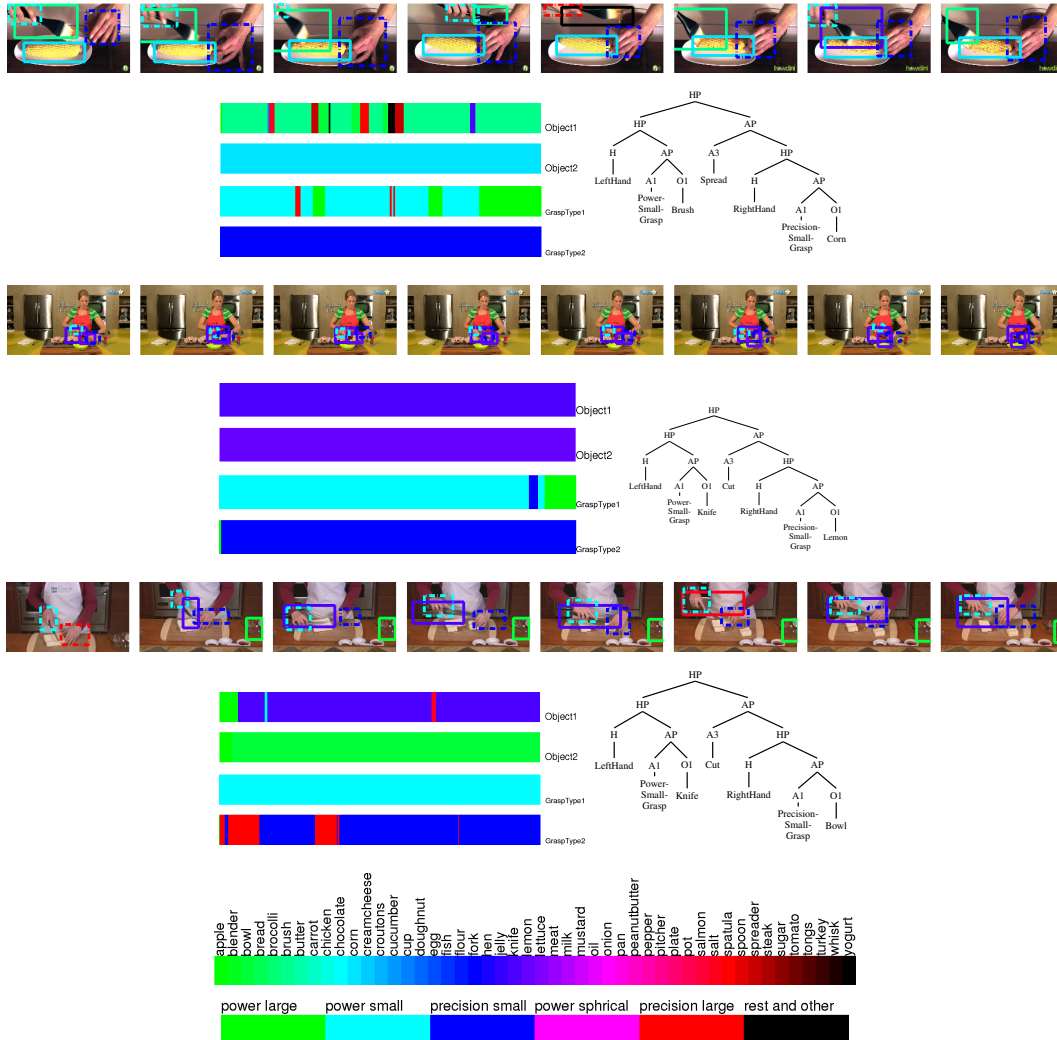


Figure 3: Upper row: input unconstrained video frames; Lower left: color coded (see llegend at the bottom) visual recognition output frame by frame along timeline; Lower right: the most likely parse tree generated for each clip.

type and objects respectively), as well as a language model for action prediction, compose the lower level of the approach. The probabilistic manipulation action grammar based Viterbi parsing module is at the higher level, and its goal is to generate atomic commands in predicate form. We conducted experiments on a cooking dataset which consists of unconstrained demonstration videos. From the performance on this challenging dataset, we can conclude that our system is able to recognize and generate action commands robustly.

We believe that the grasp type is an essential component for fine grain manipulation action analysis. In future work we will (1) further extend the list of grasping types to have a finer categorization; (2) investigate the possibility of using the grasp type as an additional feature for action recognition; (3) automatically segment a long demonstration video into action clips.

**Acknowledgements:** This research was funded in part by the support of the European Union under the Cognitive Systems program (project POETICON++), the National Science Foundation under INSPIRE grant SMA 1248056, and support from the US Army, Grant W911NF-14-1-0384 under the Project: Shared Perception, Cognition and Reasoning for Autonomy.













Snapshot	Ground Truth Commands	Learned Commands	Snapshot	Ground Truth Commands	Learned Commands
	Grasp-PoS(LH, Knife) Grasp-PrS(RH, Tofu) Action-Cut(Knife, Tofu)	Grasp-PoS(LH, Knife) Grasp-PrS(RH, <b>Bowl</b> ) Action-Cut(Knife, <b>Bowl</b> )		Grasp-PoS(LH, Blender) Grasp-PrL(RH, Bowl) Action-Blend(Blender, Bowl)	Grasp-PoS(LH, <b>Bowl</b> ) Grasp-PrL(RH, Bowl) Action- <b>Pour</b> ( <b>Bowl</b> , Bowl)
	Grasp-PoS(LH, Tongs) Action-Grip(Tongs, Chicken)	Grasp-PoS(LH, <b>Chicken</b> ) Action- <b>Cut</b> ( <b>Chicken</b> , Chicken)		Grasp-PoS(LH, Brush) Grasp-PrS(RH, Corn) Action-Spread(Brush, Corn)	Grasp-PoS(LH, Brush) Grasp-PrS(RH, Corn) Action-Spread(Brush, Corn)
	Grasp-PoS(LH, Tongs) Action-Grip(Tongs, Steak)	Grasp-PoS(LH, Tongs) Action-Grip(Tongs, Steak)		Grasp-PoS(LH, Spreader) Grasp-PrL(RH, Bread) Action-Spread(Spreader, Bread)	Grasp-PoS(LH, Spreader) Grasp-PrL(RH, <b>Bowl</b> ) Action-Spread(Spreader, <b>Bowl</b> )
	Grasp-PoS(LH, Mustard) Grasp-PrS(RH, Bread) Action-Spread(Mustard, Bread)	Grasp-PoS(LH, Mustard) Grasp-PrS(RH, Bread) Action-Spread(Mustard, Bread)		Grasp-PoS(LH, Spatula) Grasp-PrS(RH, Bowl) Action-Stir(Spatula, Bowl)	Grasp-PoS(LH, Spatula) Grasp-PrS(RH, Bowl) Action-Stir(Spatula, Bowl)
	Grasp-PoS(LH, Pepper) Grasp-PoS(RH, Pepper) Action-Sprinkle(Pepper, Bowl)	Grasp-PoS(LH, Pepper) Grasp-PoS(RH, Pepper) Action-Sprinkle(Pepper, <b>Pepper</b> )		Grasp-PoS(LH, Knife) Grasp-PrS(RH, Lemon) Action-Cut(Knife, Lemon)	Grasp-PoS(LH, Knife) Grasp-PrS(RH, Lemon) Action-Cut(Knife, Lemon)
	Grasp-PoS(LH, Knife) Grasp-PrS(RH, Broccoli) Action-Cut(Knife, Broccoli)	Grasp-PoS(LH, Knife) Grasp- <b>PoL</b> (RH, Broccoli) Action-Cut(Knife, Broccoli)		Grasp-PoS(LH, Whisk) Grasp-PrL(RH, Bowl) Action-Stir(Whisk, Bowl)	Grasp-PoS(LH, Whisk) Grasp-PrL(RH, Bowl) Action-Stir(Whisk, Bowl)
Overall Recognition Accuracy	Object: 79% Grasping type: 91% Action: 83%	Overall percentage of correct commands: 68%			

Table 2: LH:LeftHand; RH: RightHand; PoS: Power-Small; PoL: Power-Large; PoP: Power-Spherical; PrS: Precision-Small; PrL: Precision-Large. Incorrect entities learned are in red.

## References

- [1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.
- [2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. ” O’Reilly Media, Inc.”, 2009.
- [3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.
- [4] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143. Association for Computational Linguistics, 1988.
- [5] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [6] Marc Jeannerod. The timing of natural prehension movements. *Journal of motor behavior*, 16(3):235–254, 1984.
- [7] Yangqing Jia. Caffe: An open source convolutional architecture for fast feature embedding. <http://caffe.berkeleyvision.org/>, 2013.
- [8] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *International Journal of Robotics Research*, page to appear, 2014.
- [9] Arpit Mittal, Andrew Zisserman, and Philip HS Torr. Hand detection using multiple proposals. In *BMVC*, pages 1–11. Citeseer, 2011.
- [10] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the 2011 British Machine Vision Conference*, pages 1–11, Dundee, UK, 2011. BMVA.
- [11] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.
- [12] Karun B. Shimoga. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266, 1996.
- [13] Y. Yang, A. Guha, C. Fermuller, and Y. Aloimonos. A cognitive system for understanding human manipulation actions. *Advances in Cognitive Systems*, 3:67–86, 2014.
- [14] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 444–454. Association for Computational Linguistics, 2011.