Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

Levine, Pastor, Krizhevsky, Quillen

Presented by Ted Moskovitz

COMS 6731, Spring 2018

Background

- Traditional robotic grasping requires a multitude of human-engineered features and meticulously planned trajectories, a stark contrast to the dynamically adaptive and flexible motions of humans and animals

- This work takes things further than they have before by combining an order-of-magnitude larger dataset and novel CNN architecture for predicting the success of a grasp with continuous servoing to create an adaptable and effective method of robotic grasping

Context

- Work on grasping can be broadly categorized as either geometry-driven or data-driven

- Geometry-based approaches involve the analysis of the shape of the object to be grasped, and design an optimal strategy in response, i.e. (Weisz & Allen, 2012), (Rodriguez et al. 2012)

- Data-based approaches take a wide variety of forms, including those that involve human supervision i.e. (Herzog et al. 2014), but all in some way rely on repeated trials or demonstrations to learn proper poses and force application

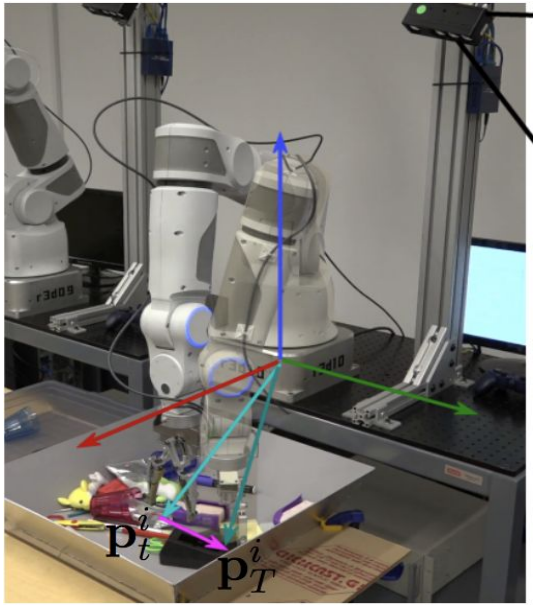
The Problem + Data Collection

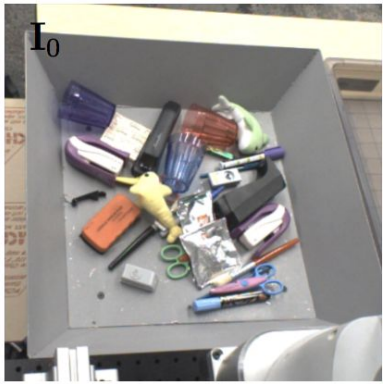
- Use grasper to pick up a variety of common office objects of varying weights, shapes, and compositions

- Use a single monocular RGB camera mounted behind grasper

- Employed 14 graspers for 2 months, ending up with >800,000 recorded grasp attempts (updated software 4x as it improved)

- Camera was mounted slightly differently for each grasper, each grasper wore out differently over time → this helped create invariance to POV + conditions





Approach: Overview

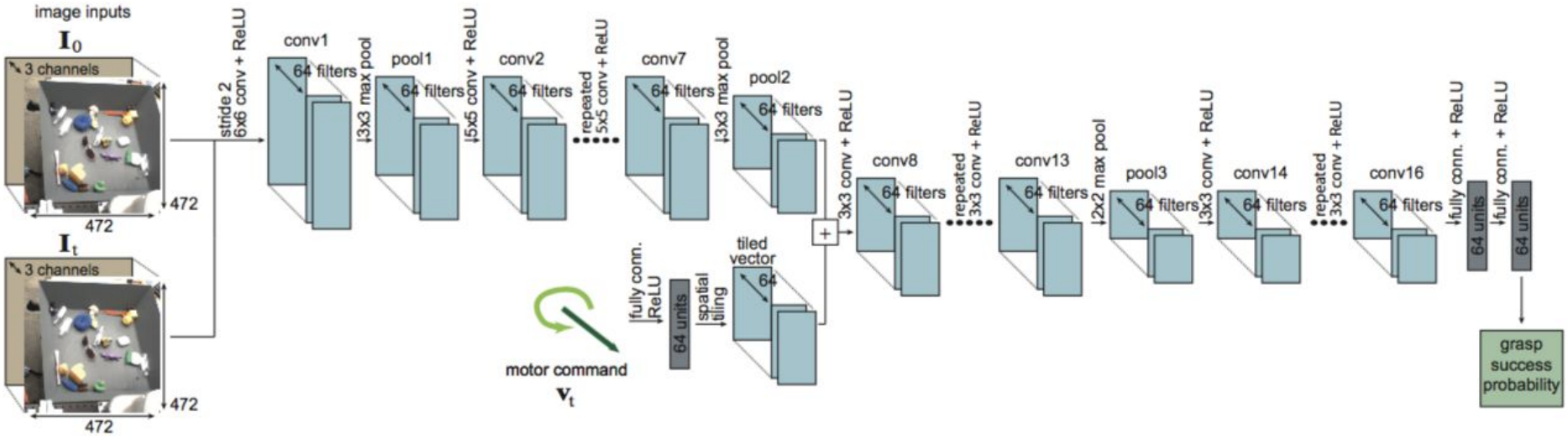
- Use a convolutional neural network (CNN) to map from the input image and current grasp candidate vector to a probability of grasp success:
$g(I_t, v_t) = Pr(success)$

- The grasp candidate vector v is determined using the Cross-Entropy Method (CEM)

- Sample several grasp candidate vectors, and pick the one with the greatest probability for success, as judged by the CNN

- Over the course of training, increased the number of allowable motions from T=2 to T=10

Approach: CNN



Approach: CEM

- Superior alternative to just randomly sampling motion vectors and picking the best

- CEM is an iterative optimization algorithm; at each iteration, it:
    ~ draws N samples from the data
    ~ fits a gaussian distribution to the best M < N of these samples
    ~ samples a new batch of N from this gaussian (here N=64, M=6)

- Used 3 iterations of CEM--picked v* from best grasp vector of final set of M=6 candidate grasps

Approach: Connection to Reinforcement Learning

- At the outside when T=2, grasp network approximates the action-value function (Q-function) of a reinforcement learning decision process with policy defined by the servoing mechanism f(I_t), and a reward of 1 for a successful grasp, and 0 otherwise

- When T>2, corresponds to the much harder problem of learning a Q-function from multiple tuples of the form $(\mathbf{I}_t, \mathbf{p}_{t+1} - \mathbf{p}_t)$

- This implies a transitive relation among states linked by actions (i.e. p1→ p2→ p3, so value of state p1 is linked to result of grasp attempt at p3)

- Authors say this is an area that they'd like to investigate more thoroughly

Approach: Servoing Mechanism

Algorithm 1 Servoing mechanism $f(\mathbf{I}_t)$

Given current image $\mathbf{I}_t$ and network g.
Infer $\mathbf{v}_t^\star$ using g and CEM.
Evaluate $p = g(\mathbf{I}_t, \emptyset)/g(\mathbf{I}_t, \mathbf{v}_t^\star)$.
if $p > 0.9$ **then**
    Output $\emptyset$, close gripper.
else if $p \leq 0.5$ **then**
    Modify $\mathbf{v}_t^\star$ to raise gripper height and execute $\mathbf{v}_t^\star$.
else
    Execute $\mathbf{v}_t^\star$.
end if

Results - Quantitative

- with/without replacement = whether object is put back in bin after being grasped

- random = untrained model

- hand-designed = uses depth images and heuristic finger positioning

- open loop = segments image into patches, selects patch with highest probability of success, and grasps there (no continuous visual feedback, but uses other part of setup)

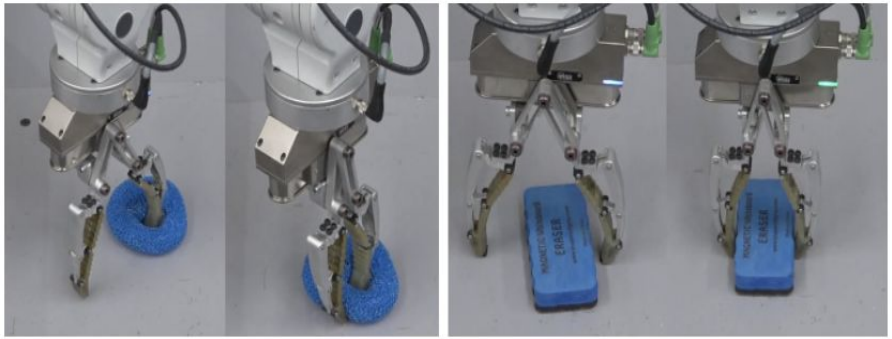
without replacement	first 10 ($N = 40$)	first 20 ($N = 80$)	first 30 ($N = 120$)
random	67.5%	70.0%	72.5%
hand-designed	32.5%	35.0%	50.8%
open loop	27.5%	38.7%	33.7%
our method	**10.0%**	**17.5%**	**17.5%**

with replacement	failure rate ($N = 100$)
random	69%
hand-designed	35%
open loop	43%
our method	**20%**

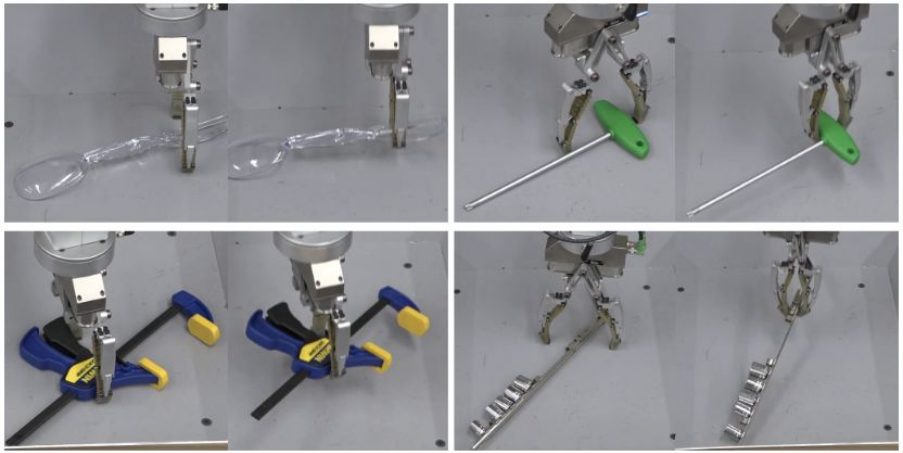
Results - Qualitative

- Video

- Developed different strategies for different types of objects:



- Also succeeded with heavy, translucent, and awkwardly shaped objects:



Implications and Takeaways

- Learned a highly capable grasping method that is invariant to camera position and small differences in hardware

- Successfully utilizes continuous servoing to avoid the need to calibrate the camera to the robot

- As a result of continuous servoing, can adopt intuitive and adaptable grasping strategies

- A novel convolutional architecture for predicting the outcome of a grasp

- A large scale data collection framework (and a much larger dataset than available previously)

Discussion Questions

- How adaptable is it really? Grasp vector is from current position to proposed end location

- There is still an issue with generalization--although robust to small differences, not transferable to other platforms

Thank you!