Article

The International Journal of
Robotics Research
2018, Vol. 37(4–5) 421–436
© The Author(s) 2017
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364917710318
journals.sagepub.com/home/ijr
SAGE

Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection

Sergey Levine[1], Peter Pastor[2], Alex Krizhevsky[1], Julian Ibarz[1] and Deirdre Quillen[1]

Abstract

We describe a learning-based approach to hand-eye coordination for robotic grasping from monocular images. To learn hand-eye coordination for grasping, we trained a large convolutional neural network to predict the probability that task-space motion of the gripper will result in successful grasps, using only monocular camera images independent of camera calibration or the current robot pose. This requires the network to observe the spatial relationship between the gripper and objects in the scene, thus learning hand-eye coordination. We then use this network to servo the gripper in real time to achieve successful grasps. We describe two large-scale experiments that we conducted on two separate robotic platforms. In the first experiment, about 800,000 grasp attempts were collected over the course of two months, using between 6 and 14 robotic manipulators at any given time, with differences in camera placement and gripper wear and tear. In the second experiment, we used a different robotic platform and 8 robots to collect a dataset consisting of over 900,000 grasp attempts. The second robotic platform was used to test transfer between robots, and the degree to which data from a different set of robots can be used to aid learning. Our experimental results demonstrate that our approach achieves effective real-time control, can successfully grasp novel objects, and corrects mistakes by continuous servoing. Our transfer experiment also illustrates that data from different robots can be combined to learn more reliable and effective grasping.

Keywords
Robotics, deep learning, neural networks

1. Introduction

When humans and animals engage in object manipulation behaviors, the interaction inherently involves a fast feedback loop between perception and action. Even complex manipulation tasks, such as extracting a single object from a cluttered bin, can be performed without first localizing the objects or mapping the scene, relying instead on continuous sensing from touch and vision (Johannson and Flanagan, 2007; Saunders and Knill, 2003). In contrast, robotic manipulation often (though not always) relies more heavily on advance planning and analysis, with relatively simple feedback, such as trajectory following, to ensure stability during execution (Srinivasa et al., 2012). Part of the reason for this is that incorporating complex sensory inputs such as vision directly into a feedback controller is exceedingly challenging. Techniques such as visual servoing (Siciliano and Khatib, 2007) perform continuous feedback on visual features, but typically require the features to be specified by hand, and both open loop perception and feedback (e.g. via visual servoing) requires manual or automatic calibration to determine the precise geometric relationship between the camera and the robot's end-effector.

In this paper, we propose a learning-based approach to hand-eye coordination for robotic grasping. Our approach is data-driven and goal-centric: our method learns to servo a robotic gripper to poses that are likely to produce successful grasps, with end-to-end training directly from image pixels to task-space gripper motion. By continuously recomputing the most promising motor commands, our method continuously integrates sensory cues from the environment, allowing it to react to perturbations and adjust the grasp to maximize the probability of success. Furthermore, the motor commands are issued in the frame of the robot's base, which is not known to the model at test time. This means that the model does not require the camera to be precisely calibrated with respect to the end-effector, but instead uses visual cues to determine the spatial relationship between the gripper and graspable objects in the scene. Our aim in designing

[1] Google
[2] X

Corresponding author:
Sergey Levine, 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA.
Email: slevine@google.com

and evaluating this approach is to understand how well a grasping system could be learned entirely from scratch, with minimal prior knowledge or manual engineering.

Our method consists of two components: a grasp success predictor, which uses a deep convolutional neural network (CNN) to determine how likely a given motion is to produce a successful grasp, and a continuous servoing mechanism that uses the CNN to continuously update the robot's motor commands. By continuously choosing the best predicted path to a successful grasp, the servoing mechanism provides the robot with fast feedback to perturbations and object motion, as well as robustness to inaccurate actuation.

The main contributions of this work are a method for learning continuous visual servoing for robotic grasping from monocular cameras, a novel convolutional neural network architecture for learning to predict the outcome of a grasp attempt, and a large-scale data collection framework for robotic grasps. We also present an extensive experimental evaluation aimed at ascertaining the effectiveness of this method, understanding its data requirements, and analyzing the potential to reuse grasping data across different types of robots. This journal article extends our earlier conference paper (Levine et al., 2016) with the addition of a second experimental evaluation on a new robotic platform (the Kuka IIWA) and evaluation of transfer learning with combined data from two different robots.

We describe two large-scale experiments that we conducted on two separate robotic platforms. In the first set of experiments, the grasp prediction CNN was trained on a dataset of about 800,000 grasp attempts, collected using a cluster of 7 degree of freedom robotic arms, shown in Figure 1. Although the hardware parameters of each robot were initially identical, each unit experienced different wear and tear over the course of data collection, interacted with different objects, and used slightly different camera poses relative to the robot base. The variety of objects provides a diverse dataset for learning grasp strategies, while the variability in camera poses provides a variety of conditions for learning continuous hand-eye coordination for the grasping task. The first experiment was aimed at evaluating the effectiveness of the proposed method, as well as comparing it to baselines and prior techniques. The dataset used in these experiments is available for download: `https://sites.google.com/site/brainrobotdata/home`

Our second set of experiments was aimed at evaluating whether grasping data collected by one type of robot could be used to improve the grasping proficiency of a different robot. In these experiments, we collected more than 900,000 additional grasp attempts using a different robotic manipulator, shown in Figure 2, with a substantially larger variety of objects. This second robotic platform was used to test whether combining data from multiple different robots results in better overall grasping proficiency.

The experiments demonstrate that our convolutional neural network grasping controller achieves a high success rate when grasping in clutter on a wide range of objects, including objects that are large, small, hard, soft, deformable, and translucent. Supplemental videos of our grasping system show that the robot employs continuous feedback to constantly adjust its grasp, accounting for motion of the objects and inaccurate actuation commands. We also compare our approach to open-loop variants to demonstrate the importance of continuous feedback, as well as a hand-engineering grasping baseline that uses manual hand-to-eye calibration and depth sensing. Our method achieves the highest success rates in our experiments. Finally, we demonstrate that we can combine data collected for two different types of robots, and use data from one robot to improve the grasping proficiency of another one.

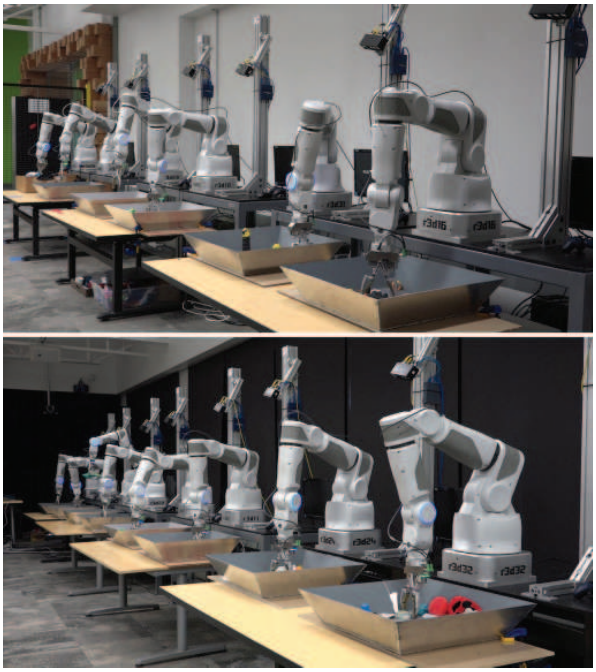

Fig. 1. Our large-scale data collection setup for the first set of experiments, consisting of 14 robotic manipulators. We collected over 800,000 grasp attempts to train the CNN grasp prediction model.

2. Related work

Robotic grasping is one of the most widely explored areas of manipulation. While a complete survey of grasping is outside the scope of this work, we refer the reader to standard surveys on the subject for a more complete treatment (Bohg et al., 2014). Broadly, grasping methods can be categorized as geometrically driven and data-driven. Geometric methods analyze the shape of a target object and plan a suitable grasp pose, based on criteria such as force closure (Weisz and Allen, 2012) or caging (Rodriguez et al., 2012). These methods typically need to understand the geometry of the scene, using depth or stereo sensors and matching of previously scanned models to observations (Goldfeder et al., 2009b).

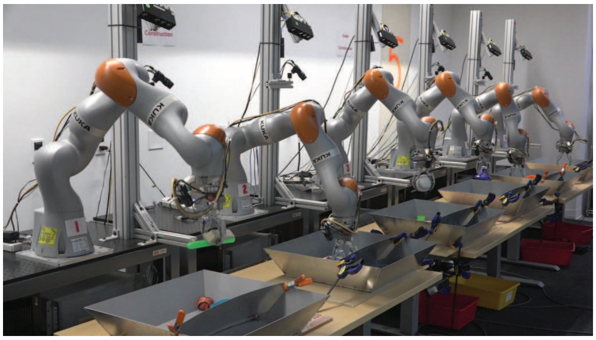

Fig. 2. Our second large-scale data collection setup, used to evaluate transfer. This setup consisted of 8 Kuka IIWA robots total, 6 shown here.

Data-driven methods take a variety of different forms, including human-supervised methods that predict grasp configurations (Ben Amor et al., 2012; Herzog et al., 2014; Kopicki et al., 2016; Lenz et al., 2015) and methods that predict finger placement from geometric criteria computed offline (Goldfeder et al., 2009a). Both types of data-driven grasp selection have recently incorporated deep learning (Gualtieri et al., 2016; Johns et al., 2016; Kappler et al., 2015; Lenz et al., 2015; Redmon and Angelova, 2015; Varley et al.). Feedback has been incorporated into grasping primarily to achieve the desired forces for force closure and other dynamic grasping criteria (Hudson et al., 2012), as well as in the form of standard servoing mechanisms, including visual servoing (described below) to servo the gripper to a pre-planned grasp pose (Kragic and Christensen, 2002). The method proposed in this work is entirely data-driven, and does not rely on any human annotation either at training or test time, in contrast to prior methods based on grasp points. Furthermore, our approach continuously adjusts the motor commands to maximize grasp success, providing continuous feedback. Comparatively little prior work has addressed direct visual feedback for grasping, most of which requires manually designed features to track the end effector (Hebert cet al., 2012; Vahrenkamp et al., 2008), with some exceptions (Arruda et al., 2016).

Our approach is most closely related to recent work on self-supervised learning of grasp poses by Pinto and Gupta (2016), as well as earlier work on learning from autonomous experimentation (Detry et al., 2011; Montesano and Lopes, 2009). Pinto and Gupta (2016). Pinto and Gupta (2016) proposed to learn a network to predict the optimal grasp orientation for a given image patch, trained with self-supervised data collected using a heuristic grasping system based on object proposals. In contrast to this prior work, our approach achieves continuous hand-eye coordination for grasping by observing the gripper and choosing the best motor command to move the gripper toward a successful grasp, rather than making open-loop predictions. As we illustrate in Section 6, this substantially improves the success rate of the method. Furthermore, our approach does not require proposals or crops of image patches and, most importantly, does not require calibration between the robot and the camera, since the closed-loop servoing mechanism can compensate for offsets due to differences in camera pose by continuously adjusting the motor commands. We trained our method using over 800,000 grasp attempts on a very large variety of objects. This is more than an order of magnitude larger than prior methods based on direct self-supervision (Pinto and Gupta, 2016) and more than double the dataset size of prior methods based on synthetic grasps from 3D scans (Kappler et al., 2015).

To collect our grasp dataset, we parallelized data collection across up to 14 separate robots. Aside from the work of Pinto and Gupta (2016), prior large-scale grasp data collection efforts have focused on collecting datasets of object scans. For example, Dex-Net used a dataset of 10,000 3D models, combined with a learning framework to acquire force closure grasps (Mahler et al., 2016), while the work of Oberlin and Tellex (2015) proposed autonomously collecting object scans using a Baxter robot. Oberlin and Tellex (2015) also proposed parallelizing data collection across multiple robots. More broadly, the ability of robotic systems to learn more quickly by pooling their collective experience has been proposed in a number of prior works, and has been referred to as collective robot learning and an instance of cloud robotics (Inaba et al., 2000; Kehoe et al., 2013, 2015; Kuffner, 2010).

Since the main purpose of our experiments is to evaluate how well a simple, entirely learning-based grasping approach can perform, we do not use any simulation data, make no use of depth sensing or wrist-mounted cameras, do not rely on a hand-designed path planner, and use no explicit representation of geometry. This stands in contrast to many successful prior grasping methods, many of which use large-scale simulated data to generate synthetic multiviewpoint depth images (Bohg et al., 2014; Gualtieri et al., 2016; Herzog et al., 2014), detect grasp affordance based on depth images (Lenz et al., 2015) that may be mounted on the wrist to establish a clearer image (Gualtieri et al., 2016), and typically use a geometric path planner to actually plan and execute a grasp (Gualtieri et al., 2016). Since our method does not use any human annotations, we can also collect a large real-world dataset entirely autonomously.

Because our method makes substantially weaker assumptions about the available human supervision (none) and the available sensing (only over-the-shoulder RGB), direct comparisons in terms of grasp success rate to values reported in prior work are not possible. The set of objects that we use for evaluation (see, for example, Figure 14) includes exceedingly difficult objects, such as transparent bottles, small round objects, deformable objects, and clutter. Discrepancies in object difficulty between our work and prior studies further complicates direct comparison of reported accuracy. The aim of our work is therefore not to illustrate which system is best, since such comparisons are impossible without standardized benchmarks, but rather examine to what degree a grasping method based entirely on learning from raw autonomously collected data can scale to complex and diverse grasp scenarios.

Another related area to our method is robotic reaching, which deals with coordination and feedback for reaching motions (Jamone et al., 2012, 2014), and visual servoing, which addresses moving a camera or end-effector to a desired pose using visual feedback (Kragic and Christensen, 2002). In contrast to our approach, visual servoing methods are typically concerned with reaching a pose relative to objects in the scene, and often (though not always) rely on manually designed or specified features for feedback control (Espiau et al., 1992; Hebert et al., 2012; Mohta et al., 2014; Vahrenkamp et al., 2008; Wilson et al., 1996). Photometric visual servoing uses a target image rather than features (Caron et al., 2013), and several visual servoing methods have been proposed that do not directly require prior calibration between the robot and camera (Hosoda and Asada, 1994; Jägersand et al., 1997; Kragic and Christensen, 2002; Yoshimi and Allen, 1994).. Several recent visual servoing methods have also used learning and computer vision techniques (Koo and Behnke, 2016; P. et al., 2016; Widmaier et al., 2016). To the best of our knowledge, no prior learning-based method has been proposed that uses visual servoing to directly move into a pose that maximizes the probability of success on a given task (such as grasping).

To predict the optimal motor commands to maximize grasp success, we use convolutional neural networks (CNNs) trained on grasp success prediction. Although the technology behind CNNs has been known for decades (LeCun and Bengio, 1995), they have achieved remarkable success in recent years on a wide range of challenging computer vision benchmarks (Krizhevsky et al., 2012), becoming the de facto standard for computer vision systems. However, applications of CNNs to robotic control problems has been less prevalent, compared to applications to passive perception tasks such as object recognition (Krizhevsky et al., 2012; Wohlhart and Lepetit, 2015), localization (Girshick et al., 2014), and segmentation (Chen et al., 2014). Several works have proposed to use CNNs for deep reinforcement learning applications, including playing video games (Mnih et al., 2015), executing simple task-space motions for visual servoing (Lampe and Riedmiller, 2013), controlling simple simulated robotic systems (Lillicrap et al., 2016; Watter et al., 2015), and performing a variety of robotic manipulation tasks (Levine et al., 2015). Many of these applications have been in simple or synthetic domains, and all of them have focused on relatively constrained environments with small datasets.

3. Overview

Our approach to learning hand-eye coordination for grasping consists of two parts. The first part is a prediction network $g(\mathbf{I}_t, \mathbf{v}_t)$ that accepts visual input $\mathbf{I}_t$ and a task-space motion command $\mathbf{v}_t$, and outputs the predicted probability that executing the command $\mathbf{v}_t$ will produce a successful grasp. The command $\mathbf{v}_t$ is specified in the frame of the robot's base and corresponds to a relative vector from the current end-effector position and a relative change in rotation about the vertical axis.

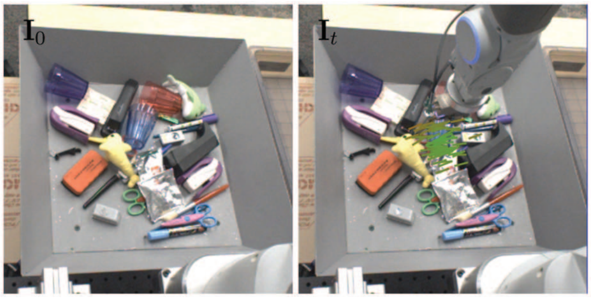

Fig. 3. Example input image pair provided to the network, overlaid with lines to indicate sampled target grasp positions. Colors indicate the probability of success: green is 1.0 and red is 0.0. The grasp positions are projected onto the image using a known calibration only for visualization. The network does not receive the projections of these poses onto the image; only offsets from the current gripper position in the frame of the robot.

The second component is a servoing function $f(\mathbf{I}_t)$ that uses the prediction network to choose the motor command $\mathbf{v}_t$ that will continuously control the robot to servo the gripper to a success grasp. We describe each of these components below: Section 4.1 formally defines the task solved by the prediction network and describes the network architecture; Section 4.2 describes how the servoing function can use the prediction network to perform continuous control.

By breaking the hand-eye coordination system into components, we can train the CNN grasp predictor using a standard supervised learning objective, and design the servoing mechanism to utilize this predictor to optimize grasp performance. The resulting method can be interpreted as a type of reinforcement learning, and we discuss this interpretation, together with the underlying assumptions, in Section 4.3. Note that the network in this scheme performs both object localization and gripper localization, subsuming both hand-eye coordination and grasp point detection. These roles are traditionally separated in standard robotic grasping systems, but we illustrate that a single model can in fact fill both roles, resulting in a simpler integrated approach. Aside from the servoing function $f(\mathbf{I}_t)$, the only other modules are an IK solver that translates end-effector commands into joint commands, and a mechanism for determining grasp success labels for training, as described in Section 5.2. Everything else is subsumed by the learned network.

In order to train our prediction network, we collected large datasets of grasp attempts using a set of similar (but not identical) robotic manipulators, shown in Figures 1 and 2. To ensure generalization of the learned prediction network, the specific parameters of each robot varied in terms of the camera pose relative to the robot, providing some invariance to camera pose (see Figure 6 for an illustration

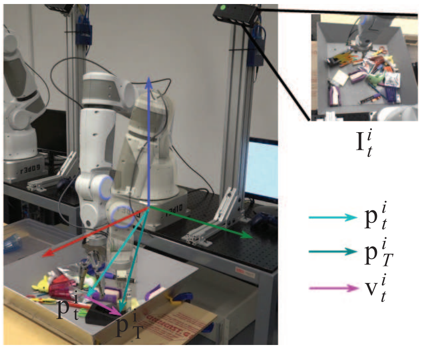

Fig. 4. Diagram of the grasp sample setup. Each grasp i consists of T time steps, with each time step corresponding to an image $\mathbf{I}_t^i$ and pose $\mathbf{p}_t^i$. The final dataset contains samples $(\mathbf{I}_t^i, \mathbf{p}_T^i - \mathbf{p}_t^i, \ell_i)$ that consist of the image, a vector from the current pose to the final pose, and the grasp success label.

of different camera poses). Furthermore, uneven wear and tear on each robot resulted in differences in the shape of the gripper fingers. Although accurately predicting optimal motion vectors in open-loop is not possible with this degree of variation, as demonstrated in our experiments, our method can correct mistakes by observing the outcomes of its past actions, achieving a high success rate even without knowledge of the precise camera calibration. Our experiments require the robot to grasp any object in a goal region, which corresponds either to a bin or, in Section 6.2, half of the bin. In principle, the same system could be used to grasp a specific object of interest simply by constraining the goal region to that object.

4. Grasping with convolutional networks and continuous servoing

In this section, we discuss each component of our approach, including a description of the neural network architecture and the servoing mechanism, and conclude with an interpretation of the method as a form of reinforcement learning, including the corresponding assumptions on the structure of the decision problem.

4.1. Grasp success prediction with convolutional neural networks

The grasp prediction network $g(\mathbf{I}_t, \mathbf{v}_t)$ is trained to predict whether a given task-space motion $\mathbf{v}_t$ will result in a successful grasp, based on the current camera observation $\mathbf{I}_t$. In order to make accurate predictions, $g(\mathbf{I}_t, \mathbf{v}_t)$ must be able to parse the current camera image, locate the gripper, and determine whether moving the gripper according to $\mathbf{v}_t$ will put it in a position where closing the fingers will pick up an object. This is a complex spatial reasoning task that requires not only the ability to parse the geometry of the scene from monocular images, but also the ability to interpret material properties and spatial relationships between objects, which strongly affect the success of a given grasp. A pair of example input images for the network is shown in Figure 3, overlaid with lines colored according to the inferred grasp success probability. Importantly, the movement vectors provided to the network are not transformed into the frame of the camera, which means that the method does not require hand-to-eye camera calibration. However, this also means that the network must itself infer the outcome of a task-space motor command by determining the orientation and position of the robot and gripper.

Data for training the CNN grasp predictor is obtained by attempting grasps using real physical robots. Each grasp consists of T time steps. At each time step, the robot records the current image $\mathbf{I}_t^i$ and the current pose $\mathbf{p}_t^i$, and then chooses a direction along which to move the gripper. At the final time step T, the robot closes the gripper and evaluates the success of the grasp (as described in Section 5.2), producing a label ℓ_i. Each grasp attempt results in T training samples, given by $(\mathbf{I}_t^i, \mathbf{p}_T^i - \mathbf{p}_t^i, \ell_i)$. That is, each sample includes the image observed at that time step, the vector from the current pose to the one that is eventually reached, and the success of the entire grasp. This process is illustrated in Figure 4. This procedure trains the network to predict whether moving a gripper along a given vector and then grasping will produce a successful grasp. Note that this differs from the standard reinforcement-learning setting, where the prediction is based on the current state and motor command, which in this case is given by $\mathbf{p}_{t+1} - \mathbf{p}_t$. We discuss the interpretation of this approach in the context of reinforcement learning in Section 4.3.

The architecture of our grasp prediction CNN is shown in Figure 5. The network takes the current image $\mathbf{I}_t$ as input, as well as an additional image $\mathbf{I}_0$ recorded before the grasp begins, which does not contain the gripper. This additional image provides an unoccluded view of the scene. The two input images are concatenated and processed by 5 convolutional layers with batch normalization (Ioffe and Szegedy, 2015), following by max pooling. After the 5th layer, we provide the vector $\mathbf{v}_t$ as input to the network. The vector is represented by 5 values: a 3D translation vector, and a sine-cosine encoding of the change in orientation of the gripper about the vertical axis. In this work, we only consider vertical pinch grasps, though extensions to other fixed-length grasp parameterizations would be straightforward. To provide this vector to the convolutional network, we pass it through one fully connected layer and replicate it over the spatial dimensions of the response map after layer 5, adding it with the output of the pooling layer. After this addition, further convolution and pooling operations are applied, as described in Figure 5. These are followed by a set of small fully connected layers that output the probability of grasp success, trained with a cross-entropy loss to match ℓ_i, causing the network to output $p(\ell_i = 1)$. The input matches are 512 × 512 pixels, and we randomly crop the images to a 472 × 472 region during training to provide for translation invariance.

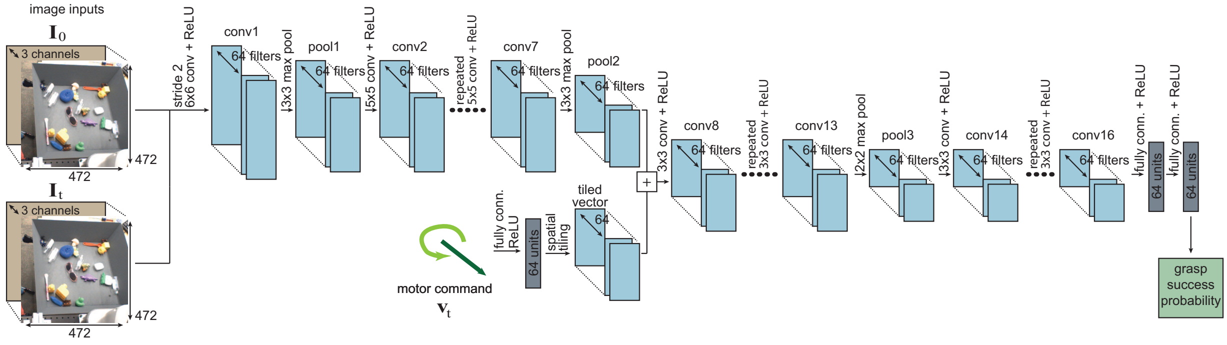

Fig. 5. The architecture of our CNN grasp predictor. The input image $\mathbf{I}_t$, as well as the pregrasp image $\mathbf{I}_0$, are fed into a 6×6 convolution with stride 2, followed by 3×3 max-pooling and six 5×5 convolutions. This is followed by a 3×3 max-pooling layer. The motor command $\mathbf{v}_t$ is processed by one fully connected layer, which is then pointwise added to each point in the response map of pool2 by tiling the output over the spatial dimensions. The result is then processed by 6 3×3 convolutions, 2×2 max-pooling, 3 more 3×3 convolutions, and two fully connected layers with 64 units, after which the network outputs the probability of a successful grasp through a sigmoid. Each convolution is followed by batch normalization.

Once trained the network $g(\mathbf{I}_t, \mathbf{v}_t)$ can predict the probability of success of a given motor command, independently of the exact camera pose. In the next section, we discuss how this grasp success predictor can be used to continuously servo the gripper to a graspable object.

4.2. Continuous servoing

In this section, we describe the servoing mechanism $f(\mathbf{I}_t)$ that uses the grasp prediction network to choose the motor commands for the robot that will maximize the probability of a successful grasp. The most basic operation for the servoing mechanism is to perform inference in the grasp predictor, to determine the motor command $\mathbf{v}_t$ given an image $\mathbf{I}_t$. The simplest way of doing this is to randomly sample a set of candidate motor commands $\mathbf{v}_t$ and then evaluate $g(\mathbf{I}_t, \mathbf{v}_t)$, taking the command with the highest probability of success. However, we can obtain better results by running a small optimization on $\mathbf{v}_t$, which we perform using the cross-entropy method (CEM) (Rubinstein and Kroese, 2004).

CEM is a simple derivative-free optimization algorithm that samples a batch of N values at each iteration, fits a Gaussian distribution to $M < N$ of these samples, and then samples a new batch of N from this Gaussian. We use $N = 64$ and $M = 6$ in our implementation, and perform three iterations of CEM to determine the best available command $\mathbf{v}_t^\star$ and thus evaluate $f(\mathbf{I}_t)$. The first iteration samples from a zero-mean Gaussian centered on the current pose of the gripper. All samples are constrained (via rejection sampling) to keep the final pose of the gripper within the workspace, and to avoid rotations of more than 180° about the vertical axis. New motor commands are issued as soon as the CEM optimization completes, and the controller runs at around 2 to 5 Hz.

One appealing property of this sampling-based approach is that we can easily impose constraints on the types of grasps that are sampled. This can be used, for example, to incorporate user commands that require the robot to grasp in a particular location, keep the robot from grasping outside of the workspace, and obey joint limits. It also allows the servoing mechanism to control the height of the gripper during each move. It is often desirable to raise the gripper above the objects in the scene to reposition it to a new location, for example when the objects move (due to contacts) or if errors due to lack of camera calibration produce motions that do not position the gripper in a favorable configuration for grasping.

We can use the predicted grasp success $p(\ell = 1)$ produced by the network to inform a heuristic for raising and lowering the gripper, as well as to choose when to stop moving and attempt a grasp. We use two heuristics in particular: first, we close the gripper whenever the network predicts that $(\mathbf{I}_t, \emptyset)$ (where $\emptyset$ corresponds to no motion) will succeed with a probability that is at least 90% of the best inferred motion $\mathbf{v}_t^\star$. The rationale behind this is to stop the grasp early if closing the gripper is nearly as likely to produce a successful grasp as moving it. The second heuristic is to raise the gripper off the table when $(\mathbf{I}_t, \emptyset)$ has a probability of success that is less than 50% of $\mathbf{v}_t^\star$. The rationale behind this choice is that, if closing the gripper now is substantially worse than moving it, the gripper is most likely not positioned in a good configuration, and a large motion will be required. Therefore, raising the gripper off the table minimizes the chance of hitting other objects that are in the way. While these heuristics are somewhat ad-hoc, we found that they were effective for successfully grasping a wide range of objects in highly cluttered situations, as discussed in Section 6. Pseudocode for the servoing mechanism $f(\mathbf{I}_t)$ is presented in Algorithm 1.

4.3. Interpretation as reinforcement learning

One interesting conceptual question raised by our approach is the relationship between training the grasp prediction

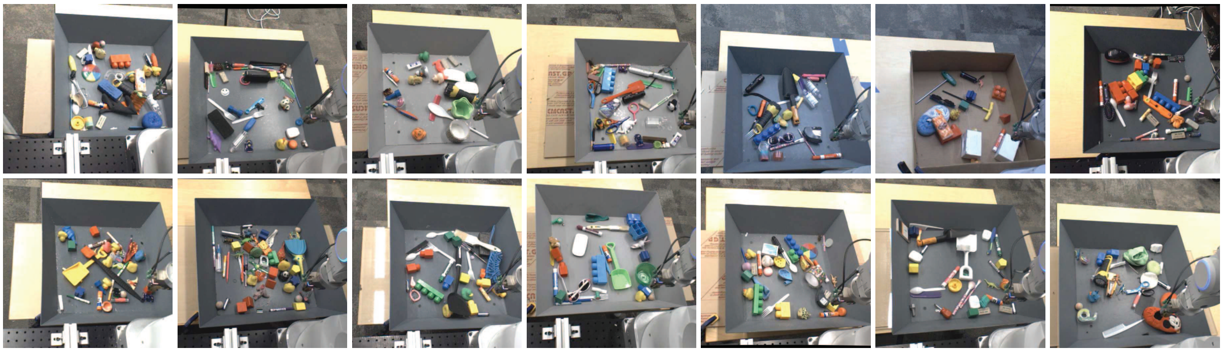

Fig. 6. Images from the cameras of each of the robots during training, with each robot holding the same joint configuration. Note the variation in the bin location, the difference in lighting conditions, the difference in pose of the camera relative to the robot, and the variety of training objects.

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
```

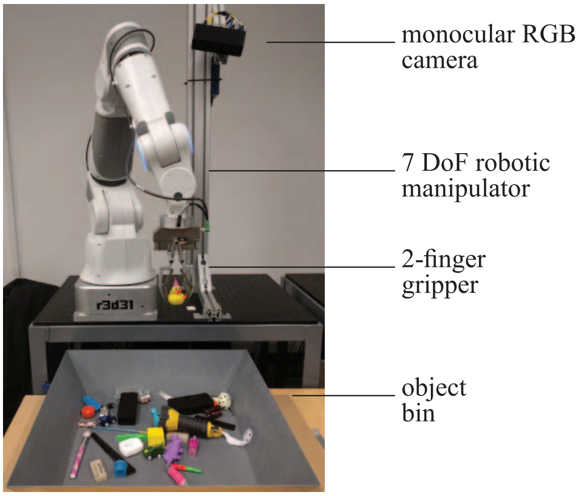

Fig. 7. Diagram of a single robotic manipulator used in our first set of experiments. Each unit consisted of a 7 degree of freedom arm with a 2-finger gripper, and a camera mounted over the shoulder of the robot. The camera recorded monocular RGB and depth images, though only the monocular RGB images were used for grasp success prediction.

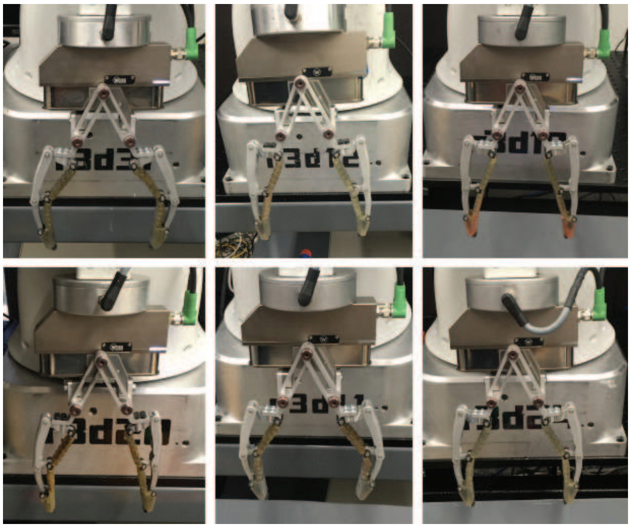

Fig. 8. The grippers of the robots used for data collection at the end of our experiments. Different robots experienced different degrees of wear and tear, resulting in significant variation in gripper appearance and geometry.

network and reinforcement learning. In the case where $T = 2$, and only one decision is made by the servoing mechanism, the grasp network can be regarded as approximating the Q-function for the policy defined by the servoing mechanism $f(\mathbf{I}_t)$, and a reward function that is 1 when the grasp succeeds and 0 otherwise. Repeatedly deploying the latest grasp network $g(\mathbf{I}_t, \mathbf{v}_t)$, collecting additional data, and refitting $g(\mathbf{I}_t, \mathbf{v}_t)$ can then be regarded as fitted Q iteration (Antos et al., 2008). However, what happens when $T > 2$? In that case, fitted Q iteration would correspond to learning to predict the final probability of success from tuples of the form $(\mathbf{I}_t, \mathbf{p}_{t+1} - \mathbf{p}_t)$, which is substantially harder, since $\mathbf{p}_{t+1} - \mathbf{p}_t$ doesn't tell us where the gripper will finish before closing (which is $\mathbf{p}_T$).

Using $\mathbf{p}_T - \mathbf{p}_t$ as the action representation in fitted Q iteration therefore implies an additional assumption on the form of the dynamics. The assumption is that the actions induce a transitive relation between states: that is, that moving from $\mathbf{p}_1$ to $\mathbf{p}_2$ and then to $\mathbf{p}_3$ is equivalent to moving from $\mathbf{p}_1$ to $\mathbf{p}_3$ directly. This assumption does not always hold in the case of grasping, since an intermediate motion might move objects in the scene, but it is a reasonable approximation that we found works quite well in practice. The major advantage of this approximation is that fitting the Q function reduces to a prediction problem, and avoids the usual instabilities associated with Q iteration, since the previous Q function does not appear in the regression. An interesting and promising direction for future work is to combine our approach with more standard reinforcement learning formulations that do consider the effects of intermediate actions. This could enable the robot, for example, to perform nonprehensile manipulations to intentionally reorient and reposition objects prior to grasping.

5. Large-scale data collection

In order to collect training data to train the prediction network $g(\mathbf{I}_t, \mathbf{v}_t)$, we used between 6 and 14 robots at any given time. An illustration of our data collection setup is shown in Figure 1. This section describes the robots used in our data collection process, as well as the data collection procedure. The dataset is available here: `https://sites.google.com/site/brainrobotdata/home`

5.1. Hardware setup

We conducted two sets of experiments to evaluate our approach. In the first set of experiments, we used a lightweight 7 degree of freedom arm, a compliant, underactuated, two-finger gripper, and a camera mounted behind the arm looking over the shoulder. An illustration of a single robot used in these experiments is shown in Figure 7. The underactuated gripper provides some degree of compliance for oddly shaped objects, at the cost of producing a loose grip that is prone to slipping. An interesting property of this gripper was uneven wear and tear over the course of data collection, which lasted several months. Images of the grippers of various robots are shown in Figure 8, illustrating the range of variation in gripper wear and geometry. Furthermore, the cameras were mounted at slightly varying angles, providing a different viewpoint for each robot. The views from the cameras of all 14 robots during data collection are shown in Figure 6.

In the second set of experiments, we used a Kuka IIWA 7 degree of freedom arm, outfitted with the same underactuated gripper and over-the-shoulder camera. An illustration of this robot is shown in Figure 9. Although the overall arrangement of the robot and camera was similar (but not identical) to the first set of experiments, variation in the workspace size, arm appearance, and dynamic properties made this an interesting domain for studying inter-robot generalization. As we will discuss in Section 6.2, we used this second robotic platform to evaluate whether combining data from multiple robots improved overall grasping performance.

5.2. Data collection

For the first set of experiments, we collected around 800,000 grasp attempts over the course of two months, using between 6 and 14 robots at any given point in time, without any manual annotation or supervision. The only human intervention in the data collection process was to replace the object in the bins in front of the robots and turn on the system. The data collection process started with random motor command selection and $T = 2$. The last command is always $\mathbf{v}_T = \emptyset$ and corresponds to closing the gripper without moving. When executing completely random motor commands, the robots were successful on 10% - 30% of the grasp attempts, depending on the objects in front of them. It is likely that this value would be lower in sparsely cluttered bins, and even lower with more complex grippers, but we found that random success rates of 10% could be achieved quite regularly in our setup. About half of the dataset was collected using random grasps, and the rest used the latest network fitted to all the data collected so far. Over the course of data collection, we updated the network 4 times, and increased the number of steps from $T = 2$ at the beginning to $T = 10$ at the end. The objects

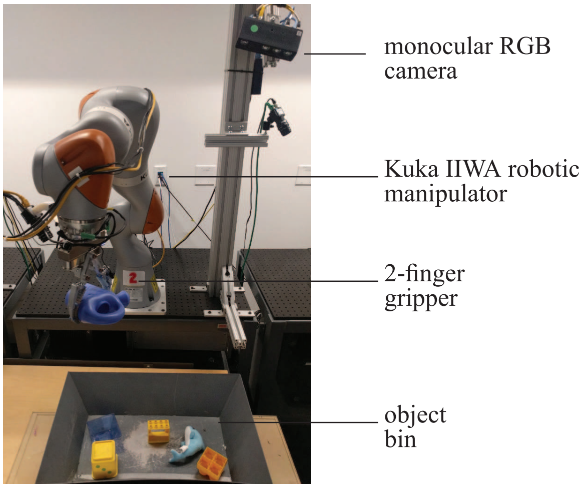

Fig. 9. Diagram of a single Kuka IIWA used in our second set of experiments. Each unit again consisted of an arm with a 2-finger gripper and over-the-shoulder camera, but with significant differences in the arm appearance and dynamics as compared to the first set of experiments.

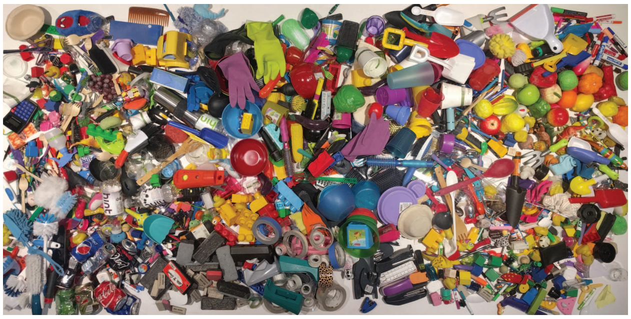

Fig. 10. Illustrations of the variety of objects used in the second set of experiments with the Kuka IIWA robots. Overall, we collected over 900,000 grasps with approximately 1100 different objects for these experiments.

for grasping were chosen among common household and office items, and ranged from a 4 to 20 cm in length along the longest axis. Some of these objects are shown in Figure 6. The objects were placed in front of the robots in metal bins with sloped sides to prevent the objects from becoming wedged into corners. The objects were periodically swapped out to increase the diversity of the training data.

For the second set of experiments, we collected over 900,000 grasp attempts over the course of about four months, using up to 9 of the Kuka IIWA robots at any given point in time, again without any manual annotation of supervision. In this experiment, we varied the objects more, with around 1100 different objects used during data collection that were swapped out regularly to maximize diversity. The objects were again chosen among common household and office items, and some examples that emphasize the diversity of these objects are shown in Figure 10. In both the first and second set of experiments, completely new, previously unseen objects were procured for testing to ensure that the experimental evaluation was testing generalization ability.

To evaluate the success of a grasp and produce a label ℓ_i, we use a combination of two mechanisms. First, we check the state of the gripper after the grasp attempt to determine whether the fingers closed completely. This simple test is effective at detecting large objects, but can miss small or thin objects. To supplement this success detector, we also use an image subtraction test, where we record an image of the scene after the grasp attempt (with the arm lifted above the workspace and out of view), and another image after attempting to drop the grasped object into the bin. If no object was grasped, these two images are usually identical. If an object was picked up, the two images will be different.

6. Experiments

To evaluate our continuous grasping system, we conducted a series of quantitative experiments with novel objects that were not seen during training. We conducted two sets of experiments. In the first set of experiments, the 7 degree of freedom arms shown in Figure 7 were used to evaluate the overall performance of our method and compare the results against baseline techniques. In the second set of experiments, the Kuka IIWA robots shown in Figure 9 were used to collect an additional grasp dataset to evaluate transfer between robots, with an evaluation of the relative performance of the system trained with data from both robots as compared to data from a single robot only.

6.1. *Experiment 1: Overall performance and comparisons*

The objects used in the evaluation for the first set of experiments are shown in Figure 11. This set of objects presents a challenging cross section of common office and household items, including objects that are heavy, such as staplers and tape dispensers, objects that are flat, such as post-it notes, as well as objects that are small, large, rigid, soft, and translucent. The goal of our evaluation for the first set of experiments was to answer the following questions: (1) does continuous servoing significantly improve grasping accuracy and success rate? (2) how well does our learning-based system perform when compared to alternative approaches? To answer question (1), we compared our approach to an open-loop method that observes the scene prior to the grasp, extracts image patches, chooses the patch with the highest probability of a successful grasp, and then uses a known camera calibration to move the gripper to that location. This method is analogous to the approach proposed by Pinto and Gupta (2016), but uses the same network architecture as our method and the same training set. We refer to this approach as “open loop”, since it does not make use of continuous visual feedback. To answer question (2), we also compared our approach to a random baseline method, as well as a hand-engineered grasping system that uses depth images and heuristic positioning of the fingers.

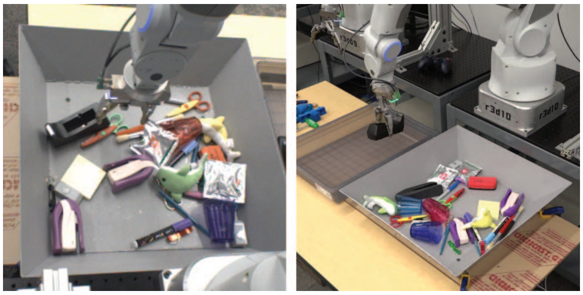

Fig. 11. Previously unseen objects used for testing (left) and the setup for grasping without replacement (right). The test set included heavy, light, flat, large, small, rigid, soft, and translucent objects.

The hand-engineered grasping system uses the depth sensor instead of the monocular camera, and required extrinsic calibration of the camera with respect to the base of the arm. For this baseline, the point clouds obtained from the depth sensor were accumulated into a voxel map, which was then segmented using graph-based segmentation into graspable object clusters. Candidate grasps were then selected for these clusters to align the fingers centrally along the longer edges of the bounding box corresponding to each detected object. This grasp configuration was then used as the target pose for a task-space controller, which was identical to the controller used for the open-loop baseline.

Note that our method was based on continuous servoing that requires fewer assumptions than either of the two alternative methods: unlike Pinto and Gupta (2016), we do not require knowledge of the camera to hand calibration, and unlike the hand-engineered grasping system, we do not require either the calibration or depth images.

We evaluated the methods using two experimental protocols. In the first protocol, the objects were placed into a bin in front of the robot, and it was allowed to grasp objects for 100 attempts, placing any grasped object back into the bin after each attempt. Grasping with replacement tests the ability of the system to pick up objects in cluttered settings, but it also allows the robot to repeatedly pick up easy objects. To address this shortcoming of the replacement condition, we also tested each system without replacement, as shown in Figure 11, by having it remove objects from a bin. For this condition, which we refer to as "without replacement", we repeated each experiment 4 times, and we report success rates on the first 10, 20, and 30 grasp attempts, which are averaged over 40, 80, and 120 grasps, respectively.

Comparisons. The results are presented in Table 1. The success rate of our continuous servoing method exceeded the baseline and prior methods in all cases. For the evaluation without replacement, our method cleared the bin completely after 30 grasps on one of the 4 attempts, and had only one object left in the other 3 attempts (which was picked up on the 31st grasp attempt in 2 of the three cases, thus clearing the bin). The hand-engineered baseline struggled to accurately resolve graspable objects in clutter, since the camera was positioned about a meter away from the table. Its performance also dropped in the non-replacement case as the bin was emptied, leaving only small, flat objects that could not be resolved by the depth camera. Many practical grasping systems use a wrist-mounted camera to address this issue (Leeper et al., 2014). In contrast, our approach did not require any special hardware modifications and used only over-the-shoulder RGB camera images. The open-loop baseline was also substantially less successful. Although it benefited from the large dataset collected by our parallelized data collection setup, which was more than an order of magnitude larger than in prior work (Pinto and Gupta, 2016), it was unable to react to objects moving when touched by the gripper and to the variability in actuation and gripper shape. The absolute performance of the open-loop method is lower than reported by Pinto and Gupta (2016). This can be attributed to differences in the setup: different objects, grippers, and clutter.

Table 1. Failure rates of each method for each evaluation condition. When evaluating without replacement, we report the failure rate on the first 10, 20, and 30 grasp attempts, averaged over 4 repetitions of the experiment. N indicates the number of grasps used to compute each value. The experiments without replacement were repeated four times.

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%**		

Table 2. Failure rates of our method for varying dataset sizes, where M specifies the number of images in the training set, and the datasets correspond roughly to the first eighth, quarter, and half of the full dataset used by our method. Note that performance continues to improve as the amount of data increases.

Without replacement	First 10 $N{=}40$	First 20 $N{=}80$	First 30 $N{=}120$
12%: $M = 182{,}249$	52.5%	45.0%	47.5%
25%: $M = 407{,}729$	30.0%	32.5%	36.7%
50%: $M = 900{,}162$	25.0%	22.5%	25.0%
100%: $M = 2{,}898{,}410$	**10.0%**	**17.5%**	**17.5%**

Evaluating data requirements. In Table 2, we evaluate the performance of our model under the no replacement condition with varying amounts of data. We trained grasp prediction models using roughly the first 12%, 25%, and 50% of the grasp attempts in our dataset, to simulate the effective performance of the model one eighth, one quarter, and one half of the way through the data collection process. Table 2 shows the size of each dataset in terms of the number of images. Note that the length of the trajectories changed over the course of data collection, increasing from $T = 2$ at the beginning to $T = 10$ at the end, so that the later datasets are substantially larger in terms of the total number of images. Furthermore, the success rate in the later grasp attempts was substantially higher, increasing from 10 to 20% in the beginning to around 70% at the end (using ϵ-greedy exploration with $\epsilon = 0.1$, meaning that one in ten decisions were taken at random). Nonetheless, these results can be informative for understanding the data requirements of the grasping task. Firstly, the results suggest that the grasp success rate continued to improve as more data was accumulated, and a high success rate (exceeding the open-loop and hand-engineered baselines) was not observed until at least halfway through the data collection process. The results also suggest that collecting additional data could further improve the accuracy of the grasping system, and we plan to experiment with larger datasets in the future.

Qualitative results. Qualitatively, our method exhibited some interesting behaviors. Figure 12 shows the grasps that were chosen for soft and hard objects. Our system preferred to grasp softer objects by embedding the finger into the center of the object, while harder objects were grasped by

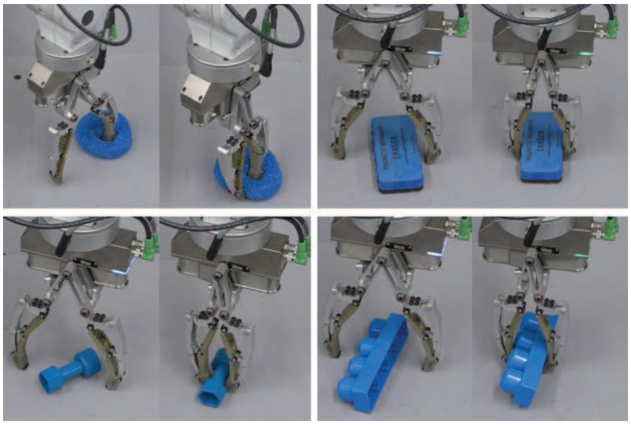

Fig. 12. Grasps chosen for objects with similar appearance but different material properties. Note that the soft sponge was grasped with a very different strategy from the hard objects.

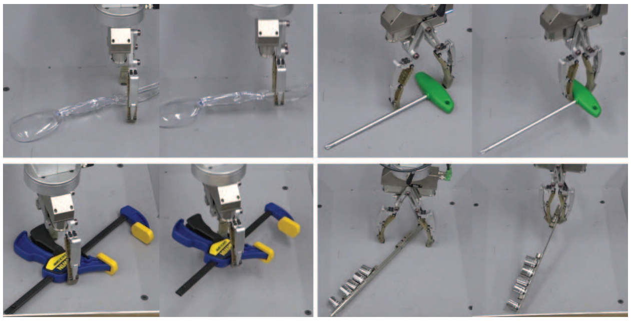

Fig. 13. Examples of difficult objects grasped by our algorithm, including objects that are translucent, awkwardly shaped, and heavy.

placing the fingers on either side. Our method was also able to grasp a variety of challenging objects, some of which are shown in Figure 13. Other interesting grasp strategies, corrections, and mistakes can be seen in our supplementary video: `https://youtu.be/cXaic_k80uM`

6.2. Experiment 2: Transfer between robots and detailed generalization analysis

The aim of our second set of experiments was to evaluate the degree to which grasping data collected for one type of robot could be repurposed to learn effective grasping skills for a second type of robot, as well as to conduct a more rigorous and detailed generalization analysis with a significantly wider and more challenging range of different objects. For these experiments, we collected an additional dataset of over 900,000 grasps using the Kuka IIWA robots shown in Figure 9, using a substantially wider range of different household and office objects, some of which are shown in Figure 10. We then trained grasp prediction networks using different amounts of data from the Kuka robots and from the first experiment. The training conditions included: (1) a network trained only on data from the robots in the first set of experiments, (2) networks trained on different amounts of data from the Kuka robots, (3) networks trained using all the data from the robots in the first experiment, as well as varying amounts of data from the Kuka robots. The third condition is meant to analyze the degree to which we can transfer data from one robot to another. This question is important for understanding whether such grasp data is indeed generalizable, or whether it is specific to a very particular hardware configuration.

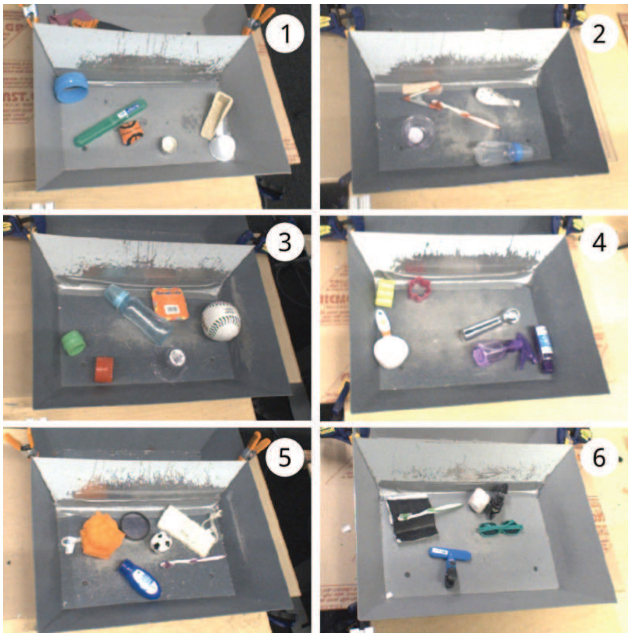

Fig. 14. Each of the test object bins used in the second set of experiments. The 6 images show a 480x320 pixel crop of the original 640x512 image fed into the network.

Experimental setup. To provide a more rigorous evaluation of generalization capability, in this experiment we used a much larger set of test objects that more exhaustively covers the range of objects that a robotic manipulator might encounter in the real world. In all, 40 different objects were used for testing, which are shown in Figure 14. Each test bin shown in the figure was presented to a different robot. None of these objects were seen in the training data. A total of 204 grasps were executed for each test bin, for a total of 1224 test grasps for each evaluation, to provide a rigorous and reliable measure of performance. The experimental protocol involved each robot picking up objects from one side of the bin and dropping them on the other side, to avoid the issues of the “with replacement” condition in the previous section, while still providing for an automated evaluation setup. The objects in this evaluation are substantially more diverse and, in many cases, more difficult than the ones tested in the first experiment. Unsurprisingly, some sets of test objects were significantly harder than others, and we present both average performance results and individual results for each bin to illustrate this.

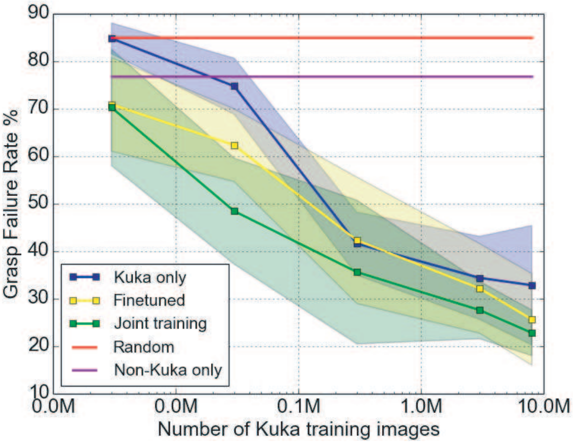

Fig. 15. Graph of transfer learning results, corresponding to the values in Table 3. The Kuka only condition indicates training with varying amounts of images from only the Kuka IIWA robots. The finetuned condition corresponds to pretraining on 2.7 million images from Experiment 1 and then finetuning on Kuka data. Joint training corresponds to jointly training with 2.7 million Experiment 1 images and varying numbers of Kuka images. Random is a random policy, and the non-Kuka only condition used only the images from Experiment 1. The results indicate that including prior data produces a small but significant improvement in final performance, with joint training producing the best results. Shaded regions correspond to standard deviations over the test bins.

Transfer results. The results for this second set of experiments are shown in Table 3, with plots in Figure 15. As discussed above, the results are presented in terms of both average failure rate, as well as the failure rate for each individual bin. First, the results indicate that even a network trained entirely on data from a different robot already performs better than chance, indicating a nontrivial amount of transfer. The results for training on only data from the Kuka robots agree with the results in Table 2: more data improves generalization. Finally, the results indicate that combining data from the Kuka robots and the robots in the first experiments does effectively improve generalization ability. When comparing conditions where an equal amount of Kuka robot data was provided to both networks, we see that networks that were also provided with three million images from the first set of experiments perform better, and the improvement in performance decreases as more Kuka data is added. The experiment also shows that training jointly on both datasets seems to perform better than training on non-Kuka data and then finetuning on Kuka, which is considered a standard practice when adapting to a new domain. This seems to be true for all ranges of data that we tested (from 3 thousand Kuka samples to 8 million). The different columns in Table 3 show the performance of each model on each bin in Figure 14. This evaluation indicates that performance changes substantially with the test objects. This is not surprising, but the results are valuable for understanding quantitative grasping evaluations. We emphasize that the objects used in these experiments are chosen from a very diverse and challenging set, including objects that are large, small, heavy, light, rigid, deformable, and transparent. It is worth noting that prior work on grasping often excludes some of these categories, especially objects that are transparent, very large, or very small. For this reason, it is difficult to compare our grasp success results directly to prior work (Gualtieri et al., 2016; Pinto and Gupta, 2016), and the results should instead be considered primarily as comparisons to the provided baselines. Our aim is not to report the highest possible success rates, but to provide a rigorous comparative evaluation.

Analysis of failures. An interesting question to ask regarding the results in the previous section is: for the difficult test bins, which objects account for most of the failures, and what do these failures suggest about the performance of our learned grasping system? A few of the example failures are illustrated in Figure 16. We found that a relatively small number of objects accounted for most of the failed grasps, and the network repeatedly performed unsuccessful grasps on these same objects, resulting in abnormally high failure rates for some bins. Recall, however, that these experiments were performed on objects that the network had never seen before during training. One interesting implication of this is that, in a real-world deployment of such learning-based grasping systems, a relatively easy and effective direction to explore for improvement is to continuously incorporate additional data as the system performs grasp attempts in the real world, rather than using a hard separation between a training and test phase. In that case, such repeated failures would quickly modify the network and avoid the same mistakes in the future, producing a robotic grasping system that continuously gets better and better from its own experience. Such an online training regime would be promising to study in future work.

The 5 most common failure cases that we observed during evaluation are illustrated in Figure 16 and listed below.

1. The particular design choices of the algorithm, especially the heuristic of adapting the height in combination with limiting the number of servoing commands to 10 accounts for some of the failure cases. Eventually, the system reaches the maximum number of allowed adaptations and is forced to attempt a grasp which likely fails. The maximum number of steps is limited partially to reduce the risk of the arm getting in singularities, which could be mitigated with a better inverse kinematics solver.
2. Over the course of the long-running data collection process, fingers wear out and were frequently replaced (see Figure 8). Some of the failed grasps can be accounted for due to the deformation of the fingers leaving room for the objects to escape despite the fingertips touching each other.

Table 3. Failure rates with Kuka IIWA robots and large-scale generalization testing (lower is better). The columns show failure rates for the different bins in Figure 14, with average failure rates and standard deviations in the leftmost column. The experimental conditions correspond, respectively, to: a random grasping baseline ("random policy"), a network trained only on the robots in the first set of experiments ("2.7m non-Kuka images"), training on different amounts of data from just the Kuka robots, training on all of the data from the first set of robots and then finetuning on varying amounts of Kuka data ("finetuned"), and training jointly on data from both robots, with varying amounts of Kuka data ("joint"). The results show that incorporating data from another robotic platform improves grasping generalization when the same amount of data from the Kuka robots is available. Training jointly seems to consistently outperform training only on Kuka data or finetuning on it on any amount of Kuka data.

Training data	Average failure rate	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6
Random policy	89.79 ± 5.8%	88.24%	94.12%	91.18%	79.90%	88.73%	96.57%
2.7m non-Kuka images	76.80 ± 12.3%	87.25%	89.22%	75.49%	84.31%	60.29%	64.22%
3k Kuka images	84.80 ± 3.6%	82.35%	91.18%	82.84%	85.29%	81.37%	85.78%
30k Kuka images	74.75 ± 6.5%	75.49%	87.25%	73.04%	69.12%	71.08%	72.55%
300k Kuka images	41.50 ± 7.3%	50.00%	49.51%	42.65%	39.78%	32.35%	35.29%
3m Kuka images	34.31 ± 9.5%	46.08%	42.16%	31.86%	30.88%	19.12%	35.78 %
8m Kuka images	32.84 ± 13.8%	52.94%	38.24%	24.02%	30.88%	12.75%	38.24%
2.7m non-Kuka images and 3k Kuka images (finetuned)	70.92 ± 10.8%	76.96%	74.51%	76.47%	73.53%	49.02%	75.00%
2.7m non-Kuka images and 30k Kuka images (finetuned)	62.34 ± 8.4%	75.49%	60.29%	62.25%	66.18%	50.00%	59.80%
2.7m non-Kuka images and 300k Kuka images (finetuned)	35.62 ± 14.6%	43.63%	34.31%	36.27%	70.49%	29.90%	39.22%
2.7m non-Kuka images and 3m Kuka images (finetuned)	32.19 ± 10.3%	50.98%	24.02%	31.37%	25.00%	25.49%	36.27%
2.7m non-Kuka images and 8m Kuka images (finetuned)	25.65 ± 10.5%	46.57%	**25.00%**	**19.12** %	22.55%	18.14%	22.55%
2.7m non-Kuka images and 3k Kuka images (joint)	70.26 ± 13.4%	67.16%	60.78%	76.96%	76.96%	50.98%	88.73%
2.7m non-Kuka images and 30k Kuka images (joint)	48.45 ± 12.2%	58.82%	55.88%	59.80%	47.55%	28.92%	39.71%
2.7m non-Kuka images and 300k Kuka images (joint)	35.62 ± 16.6%	60.29%	48.04%	37.75%	29.41%	15.69%	22.55%
2.7m non-Kuka images and 3m Kuka images (joint)	27.61 ± 6.6%	30.39%	35.78%	31.37%	27.94%	23.04%	17.16%
2.7m non-Kuka images and 8m Kuka images (joint)	**22.82 ± 5.3%**	**30.39%**	27.12%	24.18%	**19.61%**	**17.65%**	**17.97%**

3. Occasionally, we observe the model failing to choose good grasps on either of two nearby objects. Rarely, we see the robot oscillating between either object, more often we observe the robot attempting a grasp right between the two objects if the objects are parts of the same object. We assume that occlusion introduced by the gripper worsens the problem.
4. Some of the failure cases we observe can potentially be attributed to the fact that our system does not have access to depth information, which is relatively easy to mitigate in future work by using a depth sensor or stereo cameras.
5. The remaining set of failure cases can be attributed to the experimental setup and the evaluation process. Objects can be dropped in such a way that picking them up is difficult, e.g. upside down, stuck in corners or along the edges of the bin.

Addressing these failure cases through a combination of better system engineering, improved sensing, and additional training is likely to further reduce the failure rate in future work.

7. Discussion and future work

We presented a method for learning hand-eye coordination for robotic grasping, using deep learning to build a grasp success prediction network, and a continuous servoing mechanism to use this network to continuously control a robotic manipulator. Our first set of experiments involved training the system with approximately 800,000 grasp attempts from 14 distinct robotic manipulators with variation in camera pose. These experiments show that our method can operate in the absence of camera calibration

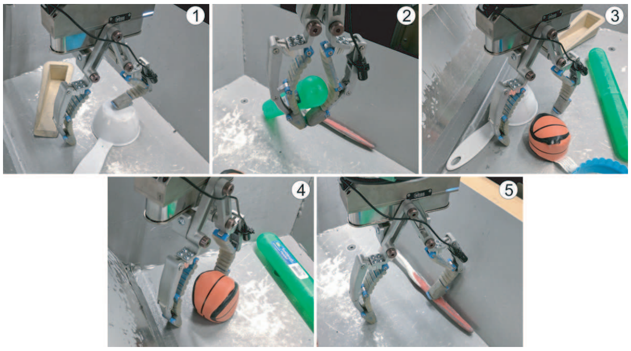

Fig. 16. Examples of some failed grasps on the Kuka IIWA test bins. A relative small number of objects accounted for a large fraction of failures. The small brown ball shown in (3) and (4) alone accounted for a large fraction of failures in bin 1, with the gripper frequently attempting to pick it up off-center, likely due to a visual cue that was too different from one seen in the training data. Additional self-supervised data collection and continuous fine-tuning at test could in principle correct such failures.

and under small variations in the hardware, camera placement, and wear and tear. Unlike most grasping and visual servoing methods, our approach does not require calibration of the camera to the robot, instead using continuous feedback to correct any errors resulting from discrepancies in calibration. Our experimental results demonstrate that our method can effectively grasp a wide range of different objects, including novel objects not seen during training. Our results also show that our method can use continuous feedback to correct mistakes and reposition the gripper in response to perturbation and movement of objects in the scene. Our second set of experiments, which involved collecting over 900,000 additional grasp attempts with a different of robots, also illustrates that pooling data from multiple robots can improve overall grasp performance, and provides a more detailed analysis of generalization with a large set of test objects.

As with all learning-based methods, our approach assumes that the data distribution during training resembles the distribution at test-time. While this assumption is reasonable for a large and diverse training set, such as the one used in this work, structural regularities during data collection can limit generalization at test time. For example, although our method exhibits some robustness to small variations in gripper shape, and some transfer occurs without any data for the second set of robots, as shown in Table 3, this transfer is quite limited, and effective training for the new robots required a considerable amount of additional data. However, once this additional data is obtained, there is considerable benefit from including the experiences of both robot types during training. A promising direction for future experiments is to analyze whether, as the system is trained on more and more robot types, generalization to new robots becomes easier and requires less experience.

The primary purpose of the experiments we presented is to study how well a grasping system might be learned entirely from scratch, with minimal prior knowledge. Of course, incorporating additional hand-designed mechanisms or priors could further improve performance, as discussed in recent work (Gualtieri et al., 2016; Mahler et al., 2016). However, our aim is to specifically study the learning component and provide a rigorous and controlled evaluation of its performance, to inform future system design. Incorporating features such as wrist-mounted depth sensing (Gualtieri et al., 2016) and pre-scanned object models (Mahler et al., 2016) is likely to further improve the performance of the method.

One of the conclusions that we might draw from our experimental evaluation is that a learning-based approach based on self-supervision requires a very significant amount of data to achieve good results. In particular, the results in Table 3 suggest that there is still room for improvement even with millions of training images, all the way up to the full dataset that consists of over a million grasps and over ten million distinct images. Although such a training set is quite time-consuming to obtain using a moderate number of robots in a laboratory setting, a grasping system deployed, for example, in a practical industrial application would likely be used across a larger number of platforms (e.g. multiple robots on an assembly line), possibly in multiple geographical locales. In this case, since all the robots can pool their experience, it would be quite feasible to quickly collect datasets of comparable size. Furthermore, the required dataset size could likely be reduced significantly by making use of more efficient classical methods to bootstrap the data collection process: our current experiments used an entirely random policy to collect the initial batch of data, with a very low success rate. Collecting this data with a more effective hand-designed grasping system would produce a more balanced dataset and substantially accelerate the early stages of learning. Using such methods primarily for dataset bootstrapping would also preserve one of the biggest strengths of the learning-based approach: data-driven grasp prediction is based entirely on the data, rather than hand-designed modeling assumptions that could introduce undesirable bias. Since any model is likely to be wrong in complex physical settings such as robotic manipulation, the data-driven approach can in principle achieve substantially improved performance when provided with enough high-quality data.

One of the most exciting aspects of this data-driven approach is the ability of the learning algorithm to discover unconventional and non-obvious grasping strategies. We observed, for example, that the system tended to adopt a different approach for grasping soft objects as opposed to hard ones. For hard objects, the fingers must be placed on either side of the object for a successful grasp. However, soft objects can be grasped simply by pinching into the object, which is most easily accomplished by placing one finger into the middle, and the other to the side. We observed this strategy for objects such as paper tissues and sponges. In future work, we plan to further explore the relationship between our self-supervised continuous grasping approach and reinforcement learning, to allow the methods to learn a wider variety of grasp strategies from large datasets of robotic experience.

At a more general level, our work explores the implications of large-scale data collection across multiple robotic platforms, demonstrating the value of this type of automatic large dataset construction for real-world robotic tasks. Although all the robots in our experiments were in a controlled laboratory environment, in the long term, this class of methods is particularly compelling for robotic systems that are deployed in the real world, and therefore are naturally exposed to a wide variety of environments, objects, lighting conditions, and wear and tear. For self-supervised tasks such as grasping, data collected and shared by robots in the real world would be the most representative of test-time inputs, and would therefore represent the best possible training data for improving the real-world performance of the system. As indicated by the analysis of failure cases in the second set of experiments, a small amount of objects account for a large fraction of grasp failures, suggesting that even a modest amount of online adaptation could quickly fix these repeated failures. So, a particularly exciting avenue for future work would to explore how our method would need to change to apply it to large-scale data collection across many deployed robots engaged in real world tasks, including grasping and other manipulation skills.

Acknowledgements

We would like to thank Kurt Konolige and Mrinal Kalakrishnan for additional engineering and insightful discussions, John-Michael Burke for running experiments, Jed Hewitt and Don Jordan for help with maintaining the robots, Max Bajracharya and Nicolas Hudson for providing us with a baseline perception pipeline, and Vincent Vanhoucke and Jeff Dean for support and organization.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was conducted at and funded by Google.

References

Antos A, Szepesvari C and Munos R (2008) Fitted Q-iteration in continuous action-space MDPs. In: *Advances in neural information processing systems*.

Arruda E, Wyatt J and Kopicki M (2016) Active vision for dexterous grasping of novel objects. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 2881–2888.

Ben Amor H, Kroemer O, Hillenbrand U, et al. (2012) Generalization of human grasping for multi-fingered robot hands. In: *IEEE/RSJ international conference on intelligent robots and systems*.

Bohg J, Morales A, Asfour T, et al. (2014) Data-driven grasp synthesis – a survey. *IEEE Transactions on Robotics* 30(2): 289–309.

Caron G, Marchand E and Mouaddib E (2013) Photometric visual servoing for omnidirectional cameras. *Autonoumous Robots* 35(2): 177–193.

Chen L, Papandreou G, Kokkinos I, et al. (2014) Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv preprint arXiv:1412.7062*.

Detry R, Kraft D, Kroemer O, et al. (2011) Learning grasp affordance densities. *Paladyn Journal of Behavioral Robotics* 2(1): 1–17.

Espiau B, Chaumette F and Rives P (1992) A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* 8(3).

Girshick R, Donahue J, Darrell T, et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE conference on computer vision and pattern recognition*.

Goldfeder C, Ciocarlie M, Dang H, et al. (2009a) The Columbia grasp database. In: *IEEE international conference on robotics and automation*.

Goldfeder C, Ciocarlie M, Peretzman J, et al. (2009b) Data-driven grasping with partial sensor data. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 1278–1283.

Gualtieri M, ten Pas A, Saenko K, et al. (2016) High precision grasp pose detection in dense clutter. *arXiv preprint arXiv:1603.01564*.

Hebert P, Hudson N, Ma J, et al. (2012) Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In: *IEEE international conference on robotics and automation*. IEEE.

Herzog A, Pastor P, Kalakrishnan M, et al. (2014) Learning of grasp selection based on shape-templates. *Autonomous Robots* 36(1–2): 51–65.

Hosoda K and Asada M (1994) Versatile visual servoing without knowledge of true Jacobian. In: *IEEE/RSJ international conference on intelligent robots and systems*.

Hudson N, Howard T, Ma J, et al. (2012) End-to-end dexterous manipulation with deliberate interactive estimation. In: *IEEE international conference on robotics and automation*.

Inaba M, Kagami S, Kanehiro F, et al. (2000) A platform for robotics research based on the remote-brained robot approach. *International Journal of Robotics Research* 19(10).

Ioffe S and Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning*.

Jägersand M, Fuentes O and Nelson RC (1997) Experimental evaluation of uncalibrated visual servoing for precision manipulation. In: *IEEE international conference on robotics and automation*.

Jamone L, Brandao M, Natale L, et al. (2014) Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching. *Robotics and Autonomous Systems* 62(4): 556–567.

Jamone L, Natale L, Metta G, et al. (2012) Autonomous online learning of reaching behavior in a humanoid robot. *International Journal of Humanoid Robotics* 9(3).

Johannson R and Flanagan R (2007) Tactile sensory control of object manipulation in humans. In: *Handbook of the Senses*.

Johns E, Leutenegger S and Davison A (2016) Deep learning a grasp function for grasping under gripper pose uncertainty. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 4461–4468.

Kappler D, Bohg B and Schaal S (2015) Leveraging big data for grasp planning. In: *IEEE international conference on robotics and automation*.

Kehoe B, Matsukawa A, Candido S, et al. (2013) Cloud-based robot grasping with the google object recognition engine. In: *IEEE international conference on robotics and automation*.

Kehoe B, Patil S, Abbeel P, et al. (2015) A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering* 12(2).

Koo S and Behnke S (2016) Focused online visual-motor coordination for a dual-arm robot manipulator. In: *IEEE international conference on robotics and automation*.

Kopicki M, Detry R, Adjigble M, et al. (2016) One-shot learning and generation of dexterous grasps for novel objects. *The International Journal of Robotics Research* 35(8): 959–976.

Kragic D and Christensen HI (2002) Survey on visual servoing for manipulation. *Computational Vision and Active Perception Laboratory* 15.

Krizhevsky A, Sutskever I and Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*.

Kuffner J (2010) Cloud-enabled humanoid robots. In: *IEEE-RAS international conference on humanoid robotics*.

Lampe T and Riedmiller M (2013) Acquiring visual servoing reaching and grasping skills using neural reinforcement learning. In: *International joint conference on neural networks*. IEEE.

LeCun Y and Bengio Y (1995) Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks* 3361(10).

Leeper A, Hsiao K, Chu E, et al. (2014) Using near-field stereo vision for robotic grasping in cluttered environments. In: *Experimental Robotics*. Heidelberg: Springer Berlin, pp. 253–267.

Lenz I, Lee H and Saxena A (2015) Deep learning for detecting robotic grasps. *The International Journal of Robotics Research* 34(4–5): 705–724.

Levine S, Finn C, Darrell T, et al. (2015) End-to-end training of deep visuomotor policies. *arXiv preprint arXiv: 1504.00702*.

Levine S, Pastor P, Krizhevsky A, et al. (2016) Learning hand-eye coordination for robotic grasping with large-scale data collection. In: *International symposium on experimental robotics*.

Lillicrap T, Hunt J, Pritzel A, et al. (2016) Continuous control with deep reinforcement learning. In: *International conference on learning representations*.

Mahler J, Pokorny F, Hou B, et al. (2016) Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In: *IEEE international conference on robotics and automation*.

Mnih V, Kavukcuoglu K, Silver D, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533.

Mohta K, Kumar V and Daniilidis K (2014) Vision based control of a quadrotor for perching on planes and lines. In: *IEEE international conference on robotics and automation*.

Montesano L and Lopes M (2009) Learning grasping affordances from local visual descriptors. In: *2009 IEEE 8th international conference on development and learning*.

Oberlin J and Tellex S (2015) Autonomously acquiring instance-based object models from experience. In: *International symposium on robotics research*.

PV, Jamone L and Bernardino A (2016) Robotic hand pose estimation based on stereo vision and GPU-enabled internal graphical simulation. *Journal of Intelligent and Robotic Systems* 83(3–4): 339–358.

Pinto L and Gupta A (2016) Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In: *IEEE international conference on robotics and automation*.

Redmon J and Angelova A (2015) Real-time grasp detection using convolutional neural networks. In: *IEEE international conference on robotics and automation*.

Rodriguez A, Mason MT and Ferry S (2012) From caging to grasping. *The International Journal of Robotics Research* 31(7): 886–900.

Rubinstein R and Kroese D (2004) *The Cross-Entropy Method*. Springer–Verlag.

Saunders J and Knill D (2003) Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental Brain Research* 152(3).

Siciliano B and Khatib O (2007) *Springer Handbook of Robotics*. New York: Springer–Verlag.

Srinivasa S, Berenson D, Cakmak M, et al. (2012) HERB 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE* 100(8): 1–19.

Vahrenkamp N, Wieland S, Azad P, et al. (2008) Visual servoing for humanoid grasping and manipulation tasks. In: *8th IEEE-RAS international conference on humanoid robots*.

Varley J, Weisz J, Weiss J and Allen P (2015) Generating Multi-Fingered Robotic Grasps via Deep Learning IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg.

Watter M, Springenberg J, Boedecker J, et al. (2015) Embed to control: A locally linear latent dynamics model for control from raw images. In: *Advances in neural information processing systems*.

Weisz J and Allen PK (2012) Pose error robust grasping from contact wrench space metrics. In: *IEEE international conference on robotics and automation*.

Widmaier F, Kappler D, Schaal S, et al. (2016) Robot arm pose estimation by pixel-wise regression of joint angles. In: *IEEE international conference on robotics and automation*.

Wilson WJ, Hulls CWW and Bell GS (1996) Relative end-effector control using Cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation* 12(5).

Wohlhart P and Lepetit V (2015) Learning descriptors for object recognition and 3D pose estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yoshimi BH and Allen PK (1994) Active, uncalibrated visual servoing. In: *IEEE international conference on robotics and automation*.