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

Sergey Levine(✉), Peter Pastor, Alex Krizhevsky, and Deirdre Quillen

Google, Menlo Park, USA
slevine@google.com

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 and independently 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. To train our network, we collected over 800,000 grasp attempts over the course of two months, using between 6 and 14 robotic manipulators at any given time, with differences in camera placement and hardware. Our experimental evaluation demonstrates that our method achieves effective real-time control, can successfully grasp novel objects, and corrects mistakes by continuous servoing.

Keywords: Deep learning · Grasping · Computer vision

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 with hardly any advance planning, relying instead on feedback from touch and vision. 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. 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.

D. Kulić et al. (eds.), *2016 International Symposium on Experimental Robotics*,
Springer Proceedings in Advanced Robotics 1, DOI 10.1007/978-3-319-50115-4_16

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

We propose a learning-based approach to hand-eye coordination, which we demonstrate on a robotic grasping task. 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, 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 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 grasp prediction CNN was trained using a dataset of over 800,000 grasp attempts, collected using a cluster of similar (but not identical) robotic manipulators, shown in Fig. 1, over the course of several months. Our experimental evaluation demonstrates 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 alternative designs 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[1].

[1] An extended version of this paper is available online [Levine et al. 2016].

2 Related Work

Robotic grasping is one of the most widely explored areas of manipulation. A complete survey of grasping is outside the scope of this work, and we refer the reader to standard surveys on the subject for a more complete treatment [Bohg et al. 2014], while in this section we primarily discuss data-driven prior grasping methods, which are the most related to the present work. Such methods take a variety of forms, including human-supervised methods that predict grasp configurations [Herzog et al. 2014, Lenz et al. 2015] and methods that predict finger placement from geometric criteria computed offline [Goldfeder et al. 2009]. Both types of data-driven grasp selection have recently incorporated deep learning [Kappler et al. 2015, Lenz et al. 2015, Redmon and Angelova 2015]. Feedback has been incorporated into grasping primarily as a way 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 [Vahrenkamp et al. 2008, Hebert et al. 2012].

Our approach is most closely related to recent work on self-supervised learning of grasp poses by [Pinto and Gupta 2016]. This prior work 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 by observing the gripper and choosing the best motor command to move the gripper toward a successful grasp, rather than making open-loop predictions. 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, which 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].

Another related area is visual servoing, which addresses moving a camera or end-effector to a desired pose using visual feedback [Kragic and Christensen 2002, Siciliano and Khatib 2007]. 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. 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).

3 Overview

Our approach to learning hand-eye coordination for grasping consists of two parts. The first part is a prediction network $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$ that accepts visual input $\boldsymbol{I}_t$ and a task-space motion command $\boldsymbol{v}_t$, and outputs the predicted probability that executing the command $\boldsymbol{v}_t$ will produce a successful grasp. The second part is a servoing function $f(\boldsymbol{I}_t)$ that uses the prediction network to continuously control the robot to servo the gripper to a success grasp. By breaking up 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. In order to train our prediction network, we collected over 800,000 grasp attempts using a set of similar (but not identical) robotic manipulators, shown in Fig. 1. 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 independence to camera calibration. 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 continuous servoing 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.

4 Grasping with CNNs 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.

4.1 Grasp Success Prediction with Convolutional Neural Networks

The grasp prediction network $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$ is trained to predict whether a given task-space motion $\boldsymbol{v}_t$ will result in a successful grasp, based on the current camera observation $\boldsymbol{I}_t$. In order to make accurate predictions, $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$ must be able to parse the current camera image, locate the gripper, and determine whether moving the gripper according to $\boldsymbol{v}_t$ 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 Fig. 2, overlaid with lines colored accordingly to the inferred grasp success probabilities. 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.

Fig. 2. Left: diagram of the grasp sample setup. Each grasp i consists of T time steps, with each time step corresponding to an image $\boldsymbol{I}_t^i$ and pose $\boldsymbol{p}_t^i$. The final dataset contains samples $(\boldsymbol{I}_t^i, \boldsymbol{p}_T^i - \boldsymbol{p}_t^i, \ell_i)$ that consist of the image, a vector from the current pose to the final pose, and the grasp success label. Right: example input image pair provided to the network, overlaid with lines to indicate sampled target grasp positions. Colors indicate their probabilities 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.

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 $\boldsymbol{I}_t^i$ and the current pose $\boldsymbol{p}_t^i$, 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 Sect. 5), producing a label ℓ_i. Each grasp attempt results in T training samples, given by $(\boldsymbol{I}_t^i, \boldsymbol{p}_T^i - \boldsymbol{p}_t^i, \ell_i)$. 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 Fig. 2. 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 $\boldsymbol{p}_{t+1} - \boldsymbol{p}_t$.

The architecture of our grasp prediction CNN is shown in Fig. 3. The network takes the current image $\boldsymbol{I}_t$ as input, as well as an additional image $\boldsymbol{I}_0$ that is recorded before the grasp begins, and 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, following by max pooling. After the 5^{th} layer, we provide the vector $\boldsymbol{v}_t$ 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.[2] 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, concatenating it with the output of the pooling

[2] In this work, we only consider vertical pinch grasps, though extensions to other grasp parameterizations would be straightforward.

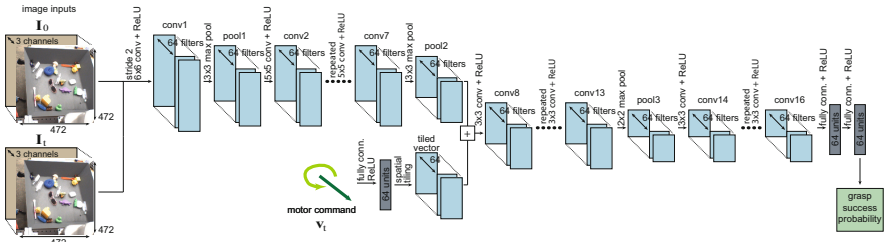


Fig. 3. The architecture of our CNN grasp predictor. The input image $\boldsymbol{I}_t$, as well as the pregrasp image $\boldsymbol{I}_0$, are fed into a 6×6 convolution with stride 2, followed by 3×3 max-pooling and 6 5×5 convolutions. This is followed by a 3×3 max-pooling layer. The motor command $\boldsymbol{v}_t$ 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 special 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.

layer. After this concatenation, further convolution and pooling operations are applied, as described in Fig. 3, 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.

Once trained the network $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$ 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 continuous servo the gripper to a graspable object.

4.2 Continuous Servoing

In this section, we describe the servoing mechanism $f(\boldsymbol{I}_t)$ that uses the grasp prediction network to choose the motor commands for the robot that will maximize the probability of a success grasp. The most basic operation for the servoing mechanism is to perform inference in the grasp predictor, in order to determine the motor command $\boldsymbol{v}_t$ given an image $\boldsymbol{I}_t$. The simplest way of doing this is to randomly sample a set of candidate motor commands $\boldsymbol{v}_t$ and then evaluate $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$, taking the command with the highest probability of success. However, we can obtain better results by running a small optimization on $\boldsymbol{v}_t$, 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 $\boldsymbol{v}_t^\star$ and thus evaluate $f(\boldsymbol{I}_t)$. 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 $(\boldsymbol{I}_t, \emptyset)$, where $\emptyset$ corresponds to no motion, will succeed with a probability that is at least 90% of the best inferred motion $\boldsymbol{v}_t^\star$. 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 $(\boldsymbol{I}_t, \emptyset)$ has a probability of success that is less than 50% of $\boldsymbol{v}_t^\star$. 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 Sect. 6. Pseudocode for the servoing mechanism $f(\boldsymbol{I}_t)$ is presented in Algorithm 1.

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

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

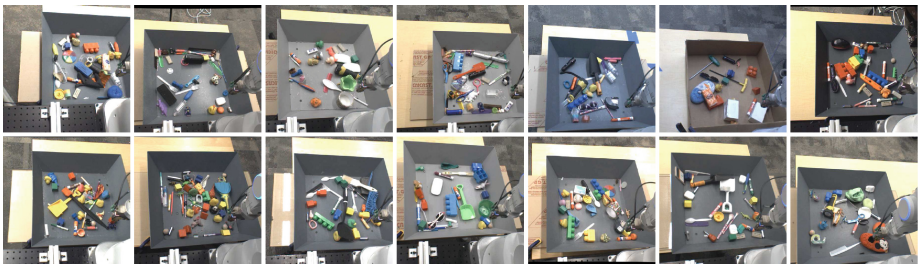
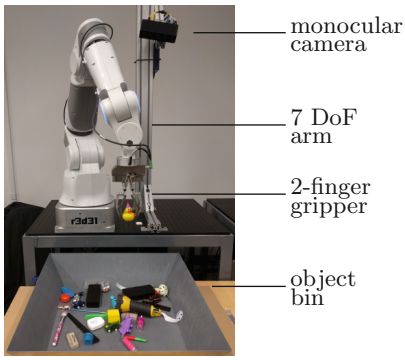


Fig. 4. Images from the cameras of each of the robots during training, with each robot holding the same joint configuration. Note the variation in the bin location, the difference in lighting conditions, the difference in pose of the camera relative to the robot, and the variety of training objects.

5 Large-Scale Data Collection



In order to collect training data to train the prediction network $g(\boldsymbol{I}_t, \boldsymbol{v}_t)$, we used between 6 and 14 robotic manipulators at any given time. A diagram of one such robot appears on the right, and an illustration of our data collection setup is shown in Fig. 1. We collected about 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 data collection process started with random motor command selection and $T = 2$, which was used to collect about half of the dataset. For the other half, the network was updated about 4 times, and the number of steps was gradually increased to $T = 10$. The last command is always $\boldsymbol{v}_T = \emptyset$ 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 current objects.

The objects were chosen to be common household and office items, and ranged from a 4 to 20 cm in length along the longest axis. Some of these are shown in Fig. 4. The objects were periodically swapped out to increase the diversity of the training data.

Grasp success was evaluated using two methods: first, we marked a grasp as successful if the position reading on the gripper was greater than 1 cm, indicating that the fingers had not closed fully. However, this method often missed thin objects, and we also included a drop test, where the robot picked up the object, recorded an image of the bin, and then dropped any object that was in the gripper. By comparing the image before and after the drop, we could determine whether any object had been picked up.

6 Experiments

To evaluate our continuous grasping system, we conducted a series of quantitative experiments with novel objects that were not seen during training. The particular objects used in our evaluation are shown in Fig. 5. 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.

Fig. 5. Previously unseen objects used for testing (left) and the setup for grasping without replacement (right). The test set included heavy, light, flat, large, small, and translucent objects.

6.1 Experimental Setup

The goal of our evaluation 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. This hand-engineered system is described further in the extended version of the paper [Levine et al. 2016]. Note that our method 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 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 Fig. 5, 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.

6.2 Comparisons

The results are presented in Fig. 6. The success rate of our method exceeded the baseline and prior methods in all cases. Without replacement, our method cleared the bin after 30 grasps on one of the 4 attempts, and had only one object left in the other 3 attempts. The hand-engineered baseline struggled to resolve graspable objects in clutter, since the camera was positioned about a meter away from the table, and 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. The open-loop baseline was also less successful. Although it benefited from the large dataset, which was more than an order of magnitude larger than in prior work [Pinto and Gupta 2016], it did not react to perturbations, movement of objects, and variability in actuation and gripper shape.

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%**		

Fig. 6. 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.

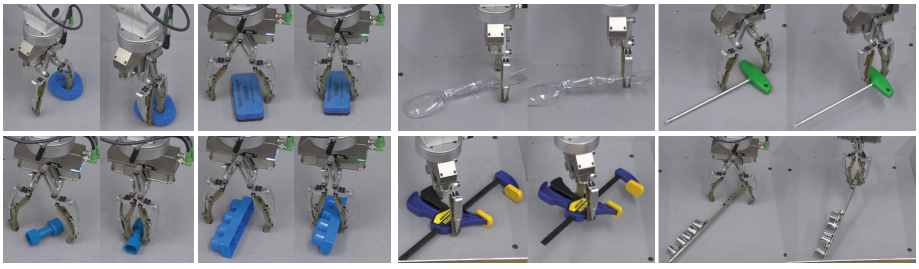


Fig. 7. Left: grasps chosen for objects with similar blue appearance but different material properties. Note that the soft sponge was grasped with a very different strategy from the hard objects. Right: examples of difficult objects grasped by our algorithm, including objects that are translucent, awkardly shaped, and heavy.

6.3 Qualitative Results

Qualitatively, our method exhibited some interesting behaviors. Figure 7 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 placing the fingers on either side. Our method was also able to grasp a variety of challenging objects, some of which are shown in Fig. 7. Other interesting grasp strategies, corrections, and mistakes can be seen in our supplementary video: https://youtu.be/cXaic_k80uM

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. By training on over 800,000 grasp attempts from 14 distinct robotic manipulators with variation in camera pose, we can achieve invariance to camera calibration and small variations in the hardware. Our approach does not require calibration of the camera to the robot, instead using continuous feedback to correct errors resulting from discrepancies in calibration. Our experiments 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.

One of the most exciting aspects of the proposed grasping method 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. In future work, we plan to further explore the relationship between our self-supervised continuous grasping approach and reinforcement learning, in order 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. 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. A particularly exciting avenue for future work is to explore how our method would need to change to apply it to large-scale data collection across a large number of deployed robots engaged in real world tasks, including grasping and other manipulation skills.

Acknowledgements. We thank Kurt Konolige and Mrinal Kalakrishnan for additional engineering and discussions, Jed Hewitt, Don Jordan, and Aaron Weiss for help with hardware, Max Bajracharya and Nicolas Hudson for the baseline perception pipeline, and Vincent Vanhoucke and Jeff Dean for support and organization.

References

Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven grasp synthesis a survey. IEEE Trans. Robot. **30**(2), 289–309 (2014)

Goldfeder, C., Ciocarlie, M., Dang, H., Allen, P.K.: The Columbia grasp database. In: IEEE International Conference on Robotics and Automation (2009)

Hebert, P., Hudson, N., Ma, J., Howard, T., Fuchs, T., Bajracharya, M., Burdick, J.: Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In: IEEE International Conference on Robotics and Automation. IEEE (2012)

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Bohg, J., Asfour, T., Schaal, S.: Learning of grasp selection based on shape-templates. Autonom. Robots **36**(1–2), 51–65 (2014)

Hudson, N., Howard, T., Ma, J., Jain, A., Bajracharya, M., Myint, S., Kuo, C., Matthies, L., Backes, P., Hebert, P.: End-to-end Dexterous manipulation with deliberate interactive estimation. In: IEEE International Conference on Robotics and Automation (2012)

Kappler, D., Bohg, B., Schaal, S.: Leveraging big data for grasp planning. In: IEEE International Conference on Robotics and Automation (2015)

Kragic, D., Christensen, H.I.: Survey on visual servoing for manipulation. Computational Vision and Active Perception Laboratory 15 (2002)

Leeper, A., Hsiao, K., Chu, E., Salisbury, J.K.: Using near-field stereo vision for robotic grasping in cluttered environments. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) Experimental Robotics. STAR, vol. 79, pp. 253–267. Springer, Heidelberg (2014)

Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. Int. J. Robot. Res. **34**(4–5), 705–724 (2015)

Levine, S., Pastor, P., Krizhevsky, A., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning, large-scale data collection. arXiv preprint (2016). arXiv:1603.02199

Pinto, L., Gupta, A.: Supersizing self-supervision: learning to grasp from 50 k tries and 700 robot hours. In: IEEE International Conference on Robotics and Automation (2016)

Redmon, J., Angelova, A.: Real-time grasp detection using convolutional neural networks. In: IEEE International Conference on Robotics and Automation (2015)

Rubinstein, R., Kroese, D.: The Cross-Entropy Method. Springer, New York (2004)

Siciliano, B., Khatib, O.: Springer Handbook of Robotics. Springer, Secaucus (2007)

Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., Dillmann, R.: Visual servoing for humanoid grasping and manipulation tasks. In: 8th IEEE-RAS International Conference on Humanoid Robots (2008)