# Real-World, Real-Time Robotic Grasping with Convolutional Neural Networks

**Conference Paper** *in* Lecture Notes in Computer Science · July 2017

DOI: 10.1007/978-3-319-64107-2_50

Joe Watson
Technische Universität Darmstadt

Fumiya Iida
University of Cambridge

# Real-World, Real-Time Robotic Grasping with Convolutional Neural Networks

Joe Watson, Josie Hughes and Fumiya Iida

Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge.
joewatson125@gmail.com, jaeh2@cam.ac.uk, fi224@cam.ac.uk

**Abstract.** Adapting to uncertain environments is a key obstacle in the development of robust robotic object manipulation systems, as there is a trade-off between the computationally expensive methods of handling the surrounding complexity, and the real-time requirement for practical operation. We investigate the use of Deep Learning to develop a real-time scheme on a physical robot. Using a Baxter Research Robot and Kinect sensor, a convolutional neural network (CNN) was trained in a supervised manner to regress grasping coordinates from RGB-D data. Compared to existing methods, regression via deep learning offered an efficient process that learnt generalised grasping features and processed the scene in real-time. The system achieved a successful grasp rate of 62% and a successful detection rate of 78% on a diverse set of physical objects across varying position and orientation, executing grasp detection in 1.8s on a CPU machine and a complete physical grasp and move in 60s on the robot.

**Keywords:** grasping, deep learning, convolution neural networks, manipulation

## 1 Introduction

Operating in an unstructured environments is a prevailing challenge in robotics. The inherent stochasticity which exists in natural settings opposes the deterministic assumption that is often the foundation of robotic design. One key challenge to achieving robotic manipulation in unstructured environments is visuomotor control for object manipulation; the process of observing and grasping an object. The key challenge here is the translation of visual perception into physical motion and the ability to robustly infer grasping points of an object to enable successful picking. Grasping points are difficult to assess due to the continuous array of viable solutions which depend on geometry, mass distribution, material and object function. The sensing and decision making should be robust to environmental disturbances. For humans, grasping objects is an instantaneous and intuitive process, but algorithmic approaches are typically search-based, either through analysing the scene or leveraging a knowledge-base.

The field of Deep Learning has yielded several significant advances for robotics, particularly in scene understanding and motor control [17]. The general neural network architecture has been shown repeatedly to be able learn general, complex relationships from data; and that their levels of performance validate the computational and data intensive requirements.

This paper investigates the use of vision-based learning for robotic grasping, specifically using Convolutional Neural Networks (CNNs) to infer grasping coordinates from vision and depth. The approach will use a Kinect sensor and will infer the grasping points from vision using regression, reproducing the work of Redmon & Angelova [14]. This approach achieved impressive results, but was never implemented on a robotic platform or tested outside the dataset. This research implements this approach in real-time using the Baxter robot platform.

### 1.1 Literature Review

Deterministic, object-orientated grasping was an early approach to robotic grasping, using CAD models and simulation to find the optimal grasping configuration based on the object and manipulator. GraspIt was devised in 2011 [13], with a large repository of 3D models of objects developed [11]. This used a deterministic, grasp-orientated approach, using an algorithm to search and extract cylindrical templates from a raw point cloud feed. This appears to work well for cluttered scenes, but is biased towards cylindrical objects and handles. Probabilistic approaches were pioneered at Cornell, initially using filter banks, search and logistic regression trained on synthetic data [6], the machine learning approach was eventually extended to the field of deep learning. Lenz [9] devised a two-stage neural network, using a sliding window to detect viable grasping rectangles and a second neural network to select the best. Although a costly search approach, it achieved impressive results, generalising well to new objects and able to sort cluttered scenes. This approach was extended to training a single convolutional neural network to regress the grasping coordinates directly [14]. Using fast GPU libraries, the system was able to operate in real-time, however this was not physically implemented on a robot. Since this research was undertaken, several new approaches have been developed. One is ‘self-supervision’, where a trained network is used to generate a larger dataset autonomously, which is then used for further training. Pinto [12] first explored this approach, using the network to classify viable grasps via the wrist camera as the arm moved around the scene, an approach similar to Lenz. Levine [10] developed self-supervision further with multiple robots training in unison, and integrating the CNN as a visual servoing controller from just a single monocular camera. Johns [7] used a CNN as a *grasp function* that learns to classify a ’score’ (discretized success rate from experience) for each location in the input image, and was trained on synthetic data.

## 2 System & Model

### 2.1 Experimental Setup

The Baxter research robot [1] has been used to demonstrate the grasping capabilities. The robot has two seven degrees of freedom arms, each with an embedded camera and parallel gripper with a travel of 35 mm. The robot is controlled using ROS (Robot Operating System) [2], an open-source software framework designed for robotic systems and provides an inverse kinematics solver for position control. A Kinect sensor mounted on top of Baxter is used to stream RGB-Depth (RGB-D) data for the surrounding environment. The experimental setup is illustrated in Figure 1.

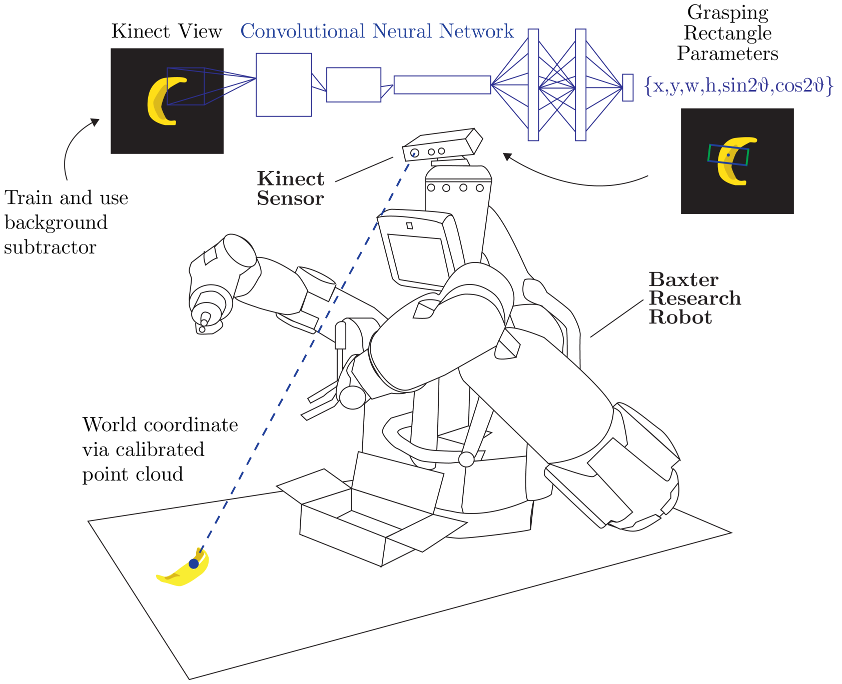
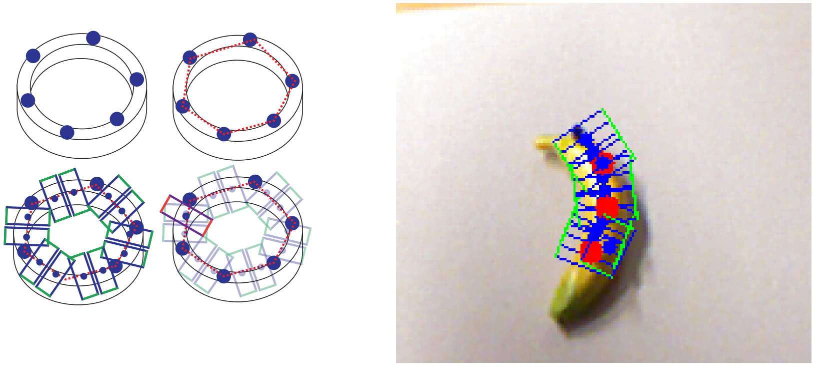

Kinect View
Convolutional Neural Network
Grasping Rectangle Parameters
{x,y,w,h,sin2ϑ,cos2ϑ}
Train and use background subtractor
Kinect Sensor
Baxter Research Robot
World coordinate via calibrated point cloud

Fig. 1: Illustration of the robotic grasping process, from processing the Kinect feed to inferring the spatial location of the grasping point.

## 2.2 Grasping procedure

The scene is preprocessed from the RGB-D stream into an RGD stream with background subtraction, using the popular Gaussian Mixture Model approach [19]. Applying the subtractor to the training data and real-time feed reduces possible variations between the two from the CNN's perspective. The image is then cropped to 224x224 and passed through the CNN, which produces a estimate of the grasping rectangle for the parallel gripper in pixel-space. These coordinates are converted to the robot's world frame of reference by using the disparity view in the Kinect's frame of reference and the geometrical transform between the two, which was found through calibration. Baxer's prepackaged inverse kinematics and position control are then used for motor control. As with the trend in previous literature, the grasping is tackled in a planar manner, with the wrist generally perpendicular to the surface. The five-dimensional parameterization of the grasping rectangle is illustrated in Figure 9. Following Redmon [14], the sine and cosine of the orientation are used as the network's output, in order to encapsulate the 180° symmetry of the rectangle.

### 2.3 Online evaluation



(a) Illustration of process

(b) System in use on a banana

Fig. 2: Marker-based method of online grasping rectangle ground truth generation for evaluation.

To properly assess the grasping estimate in the real world without extensive hand-labelling, a novel automatic approach was devised using markers. With the planar approach, viable grasping rectangles can be assumed to exist perpendicular to set of paths (such as ridges, handles, etc). By placing markers and detecting them using computer vision, these paths could be piecewise-linearly approximated and used to generate a viable set of approximate ground-truth grasping rectangles. The closest match was then used to evaluate the CNN's output. An illustration of this process, and example of it in action, can be seen in Figure (2).

### 2.4 Deep Learning

The convolutional neural network is based on the model developed by Redmon [14] but trained from scratch using the Caffe library [5] rather than the CUDNN library [4] used in their work. The architecture takes advantage of Transfer Learning [3] to speed up training of the convolutional layers, so it is trained using a variant of the popular AlexNet network [8] pretrained on the large ImageNet image classification dataset. The fully connected layers were adapted to regress to the six-dimensional grasping rectangle parameters, using the batch-mean squared error cost function and initialised using Caffe's Xavier distribution. To stabilise regression, the pixel intensities of the input were scaled to the range [0,1], and the pixel labels were scaled down by 224, making them a fraction of the input dimension. Like Redmon, to make use of the pretrained three channel input and the four channel RGB-D data, the blue channel is arbitrarily removed and RGD data is passed to the network input.

The Cornell grasping dataset contains 280 everyday items hand-labelled with a variable number of correct and incorrect grasping rectangles. Only the single rectangle estimator from Redmon's work was replicated, and was trained on the first positive label from the dataset to avoid averaging of the positive labels. The data was randomly augmented x27 times via translation and rotation, to reduce overfitting.

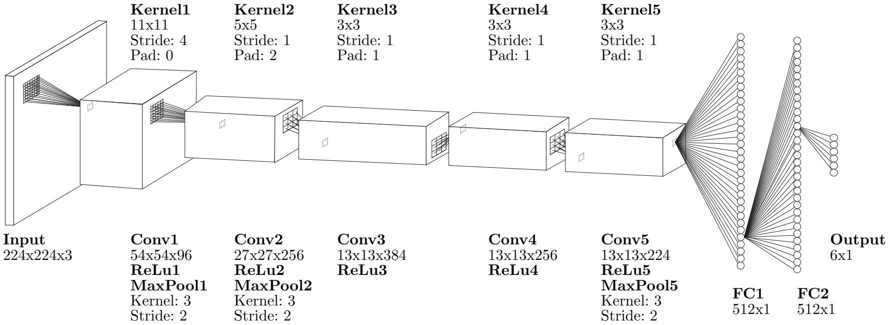


Fig. 3: Schematic of AlexNet-based CNN architecture for estimating a single grasp

The network was trained to 26 epochs over 42 hours on an AWS EC2 2.2xlarge server. The base learning rate was 0.0005, and scaled by 0.8 every three epochs. Momentum and weight decay were used, with weights of 0.8 and 0.001 respectively. Dropout was used [16], with probability of 0.5 applied to the fully connected units, to reduce overfitting.

## 3 Results

### 3.1 Deep Learning

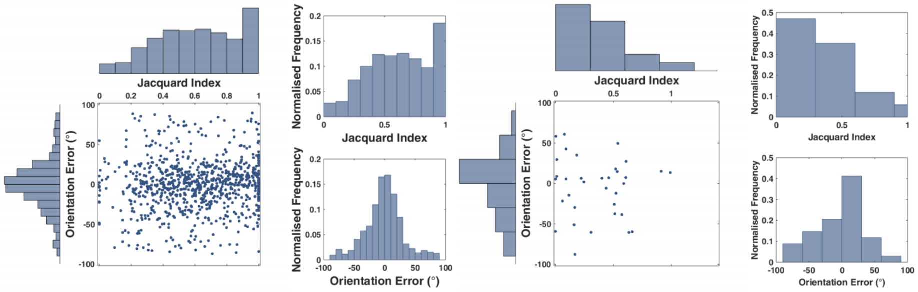
The network was assessed by single-fold cross-validation. As with Lenz and Redmon, the rectangles are evaluated using the Jacquard Index between the areas of the estimate and ground-truth rectangle, which incorporates an allowable spatial variation in the position.

$$\text{For A and B, Jacquard Index } \; J(A,B)|\& = \frac{|A \cap B|}{|A \cup B|} \; \text{ so } \; 0 \leq J(A,B) \leq 1 \qquad (1)$$

Lenz outlined a rectangle metric [9] for evaluating viable grasp, where by the Jacquard Index should be greater than 0.25 and the orientation error less than 30° in comparison to the ground truth.

| Dataset | Jacquard Index | Orientation Error (°) | | Rectangle Metric |
|---|---|---|---|---|
| | Mean | Mean | St. Dev. | Success (%) |
| Training | 0.61 | -0.95 | 31.4 | 81 |
| Testing | 0.35 | -4.4 | 36.9 | 50 |

Table 1: Statistics of CNN evaluation



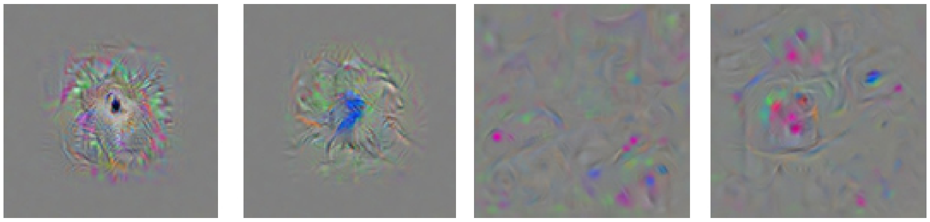
(a) Training data

(b) Testing data

Fig. 4: Scatter plot and marginal histograms of training results

### 3.2 Visualisation

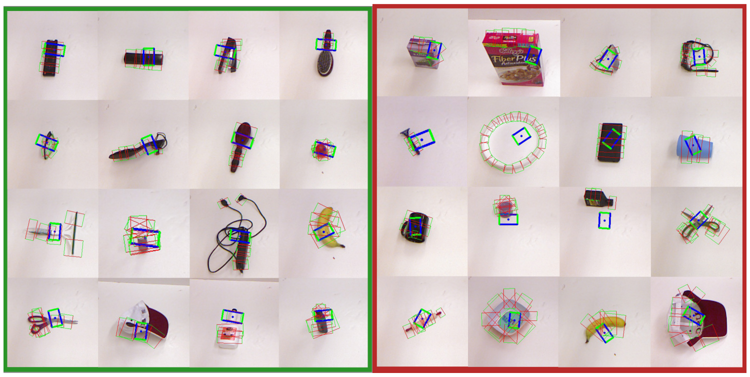
To understand what the networks' learning process and ensure it behaves as expected, the units of the last fully connected layer were visualised using the Deep Visualisation Toolbox [18] and compared to the ImageNet-trained classification model used as a base for transfer learning. The visualisation works by optimising white noise at the input to maximise the response of the intended unit. A comparison between responses can be seen in Figure (6), and shows that the network has learnt subtle grasping features - typically orientated features and edges. To aid generalisation, each unit has also learnt a variety of features across the input space. The difference between regression and classification is also clear, with the classification having a strong, textural response while with regression the activations are far more subtle.



(a) Image classification

(b) Grasp regression

Fig. 6: Visualisations of the activations of the final fully connected layer units for two AlexNet-based architectures.



(a) Correct

(b) Incorrect

Fig. 5: Visual results on training data

### 3.3 Physical Grasping

For a physical grasping system, it is difficult to evaluate performance beyond pass or fail. For this system, it felt important to separate the performance of the grasp detection subsystem (incorporating the image processing and CNN) and the kinematic subsystem (involving the calibrated point cloud, inverse kinematics and gripper). Assessing the grasp detection would involve generating a new ground truth data set for the experimental set-up, and labelling each object consistently and thoroughly would be immensely time-consuming work. The online evaluation method outlined in Section 2.3 was used to efficiently evaluate the CNN in a real-world setting. The physical system was assessed in a pass/fail manner, across multiple varied everyday objects in a variety of positions and orientations. A workspace was divided into a 3x3 grid 45 cm square, with each object randomly placed within each cell at two perpendicular orientations. Rotationally symmetric objects were tested in only one orientation.

The Jacquard index and orientation error are given in Table 2 for the live datasets in addition to the training and testing datasets. The success rate for determining viable grasping rectangles and for physical grasping of objects are given in Figure 7 and Figure 8.

Table 2: Statistics of physical evaluation

| Dataset | Jacquard Index | Orientation Error (°) | | Rectangle Metric Success (%) |
|---|---|---|---|---|
| | Mean | Mean | St. Dev. | |
| Live | 0.63 | 15.7 | 20 | 78 |
| Training | 0.61 | -0.95 | 31.4 | 81 |
| Testing | 0.25 | -4.4 | 36.9 | 50 |

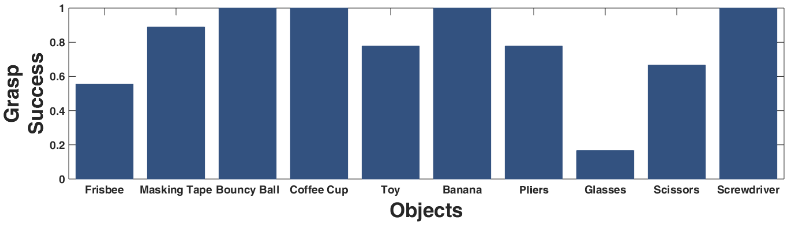


Fig. 7: Success rate of viable grasping rectangle estimates over objects for physical grasping

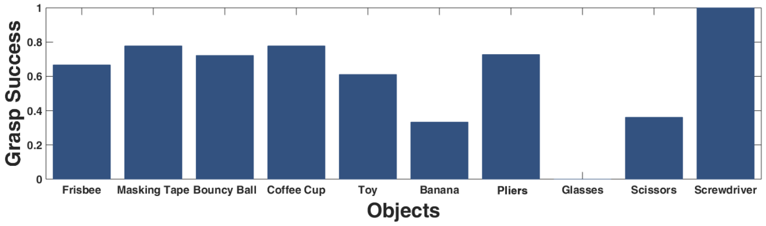


Fig. 8: Success rate for physical grasping over objects

The grasping system was implemented successfully, with each grasp executing within around 60 seconds. Figure 8 shows that the system is reasonably consistent, with the physical aspect only reducing the expected success rate (based on the rectangle estimates) by a small amount (comparing Figures 7 and 8). The weakness to the frisbee, glasses and scissors was down to the CNN, while the weakness due to the physical system was primarily seen in the ball and banana. The ball requires precise positioning, which Baxter and the inverse kinematics cannot provide. Despite a correct rectangle and only a small placement error, the grasp was not successful. Interestingly, the interpolation problem can be observed with the pliers, but the handles are close enough to still enable a good grasp. A link to videos of the system in use can be found in the Appendix.

The use of Background Subtraction became a significant issue in the system, with background variation increasing over time and affecting performance of the network. Shadows were also an issue, as they interpreted as part of the object.

## 4 Discussion & Conclusion

The results offer an initial demonstration that a robotic grasping system which uses a regression-based CNN can be used as a functional real-time grasping point estimator. The physical implementation has successfully integrated a range of robotic paradigms. The system was found to successfully grasp a range of items, across orientation and workspace location. The novel method of ground truth approximation for evaluation, using markers and image processing, was found to produce results consistent with the actual ground truth assessment.

## Acknowledgements

Thanks to Lenz [9] & Saxena [15] for producing training dataset used, Pinot & Gupta [10] for helpful discussion, and Dr Ankur Handa for support and encouragement.

## References

1. Baxter — redefining robotics and manufacturing — rethink robotics. http://www.rethinkrobotics.com/baxter/, (Accessed on 05/19/2016)
2. Ros.org — powering the world's robots. http://www.ros.org/, (Accessed on 05/22/2016)
3. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. In: Guyon, I., Dror, G., Lemaire, V., Taylor, G.W., Silver, D.L. (eds.) ICML Unsupervised and Transfer Learning. JMLR Proceedings, vol. 27, pp. 17–36. JMLR.org (2012), http://dblp.uni-trier.de/db/journals/jmlr/jmlrp27.htmlBengio12
4. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E.: cudnn: Efficient primitives for deep learning. CoRR abs/1410.0759 (2014), http://arxiv.org/abs/1410.0759
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., Eecs, U.C.B.: Caffe: Convolutional Architecture for Fast Feature Embedding (2014)
6. Jiang, Y., Amend, J.R., Lipson, H., Saxena, A.: Learning hardware agnostic grasps for a universal jamming gripper. In: ICRA. pp. 2385–2391. IEEE (2012)
7. Johns, E., Leutenegger, S., Davison, A.J.: Deep learning a grasp function for grasping under gripper pose uncertainty. CoRR abs/1608.02239 (2016), http://arxiv.org/abs/1608.02239
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems pp. 1–9 (2012)
9. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. The International Journal of Robotics Research 34(4-5), 705–724 (2015)
10. Levine, S., Pastor, P., Krizhevsky, A., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. CoRR abs/1603.02199 (2016), http://arxiv.org/abs/1603.02199
11. Pas, A., Platt, R.: Localizing handle-like grasp affordances in 3D point clouds. International Symposium on Experimental Robotics (2014)
12. Pinto, L., Gupta, A.: Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours (2015), http://arxiv.org/abs/1509.06825
13. Popovic, M., Kootstra, G., Jorgensen, J.A., Kragic, D., Kruger, N., Jørgensen, J.A., Krueger, N.: Grasping unknown objects using an Early Cognitive Vision system for general scene understanding. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on pp. 987–994 (2011)
14. Redmon, J., Angelova, A.: Real-Time Grasp Detection Using Convolutional Neural Networks. Proceedings - IEEE International Conference on Robotics and Automation 36(2), 1316–1322 (2015)

15. Saxena, A., Wong, L., Quigley, M., Ng, A.Y.: A Vision-based System for Grasping Novel Objects in Cluttered Environments
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958 (Jan 2014), http://dl.acm.org/citation.cfm?id=2627435.2670313
17. Tai, L., Liu, M.: Deep-learning in mobile robotics - from perception to control systems: A survey on why and why not. CoRR abs/1612.07139 (2016), http://arxiv.org/abs/1612.07139
18. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. In: Deep Learning Workshop, International Conference on Machine Learning (ICML) (2015)
19. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02. pp. 28–31. ICPR '04, IEEE Computer Society, Washington, DC, USA (2004), http://dx.doi.org/10.1109/ICPR.2004.479

## 5 Appendix

Videos of the grasping system may be found at `tiny.cc/birlDeepGrasp`. Full training and testing results may be found at `tiny.cc/birlTraining` and `tiny.cc/birlTesting`. The ROS-related code may be found at `tiny.cc/birlGraspCode`.

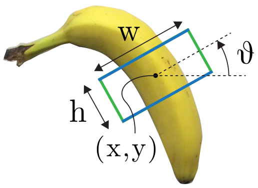


Fig. 9: Illustration of parameterization of grasping rectangle