Learning from Successes and Failures to Grasp Objects with a Vacuum Gripper

Luca Monorchio, Daniele Evangelista, Marco Imperoli, and Alberto Pretto

***Abstract*— In this work we present an empirical approach for solving the grasp synthesis problem for anthropomorphic robots equipped with vacuum grippers. Our approach exploits a self-supervised, data-driven learning approach to estimate a suitable grasp for known and unknown objects. We employ a Convolutional Neural Network (CNN) that directly infers the grasping points and the approach angles from RGB-D images as a regression problem. In particular, we split the image into a cell grid where the CNN provides, for each cell, an estimate of a grasp along with a confidence score. We collected a training dataset composed by 4000 grasping attempts by means of an automatic trial-and-error procedure, and we trained end-to-end the CNN directly on both the grasping successes and failures. We report a set of preliminary experiments performed by using known (i.e., object included in the training dataset) and unknown objects, showing that our system is able to effectively learn good grasping configurations.**

I. Introduction

Grasping is an essential task in several robotic applications, e.g. from the classical bin-picking and pick-and-place applications, to collaborative manufacturing and human robot interaction. However, reliable grasping still remains an open problem that involves challenges in several fields such as perception, planning and control. One of the most important topic within the grasping research area is grasp synthesis, that is the problem of selecting the grasping points and the position of the gripper that maximizes some grasp quality metric. Standard approaches rely on analytic formulations, where the involved kinematics and dynamics are considered in determining grasps. Due to the high number of involved constraints, often these approaches exploits some assumptions (e.g., simplified contact configurations and rigid-body assumption) in order to simply the formulation [14]: The reduction of the grasp solution space is actually one of the main challenge for such approaches.

In order to avoid the computational complexity of analytical formulations, several empirical or *data-driven* approaches and tools were recently introduced to solve the grasping problem [2], [14]. Early data-driven approaches typically sampled a discrete number of grasp candidates using ad-hoc simulation tools (e.g., [9]), the candidates were then ranked by using classical metrics: unfortunately, several works have later highlighted that often such metrics are not good predictors for grasp success in real world applications [2]. More recent approaches addressed the data-driven grasp synthesis problem by exploiting supervised machine learning approaches (among others, [15], [5], [11]), achieving promising results if proper labeled training data is available. Training data can be manually labeled [3], [5], [11], synthetically generated and labeled [15], [4], [7], or self-labeled [6], [10].

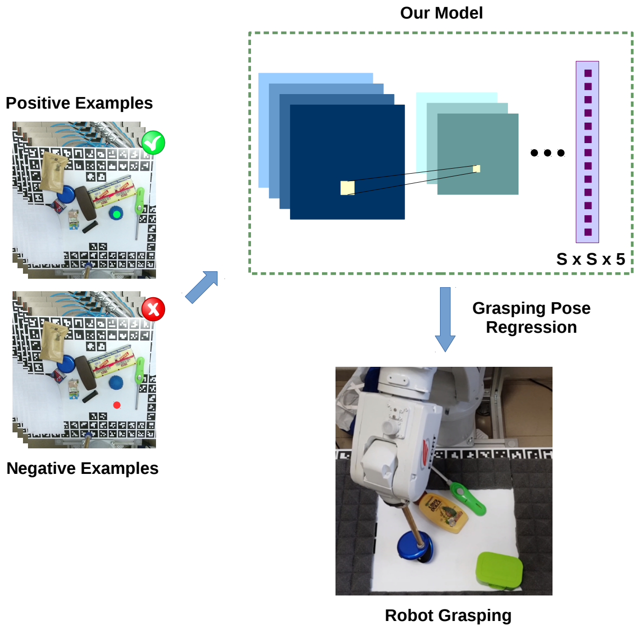

Fig. 1: We face the grasp synthesis problem by means of a self-supervised, data-driven procedure. An anthropomorphic robot self collects and labels the data needed to train our model, which is based on a CNN that predicts grasps by means of a set of regressions over a $S \times S$ grid. We train the CNN taking into account both positive and negative samples.

Following the machine learning trends, most of these empirical approaches are based on deep learning architectures: the grasp synthesis, typically for parallel-jaw or multi finger grippers, is often faced as a classification problem[1] where a CNN receives as input candidate sub-regions of an RGB image [10], a depth map [7] or an RGB-D image [5], [11] of the working area.

In this work, we propose a novel self-supervised, data-driven approach for anthropomorphic robot manipulators equipped with vacuum grippers and a RGB-D camera (Fig. 1).

Given an RGB image $\mathcal{I}$ and the corresponding depth map $\mathcal{D}$ of the working area populated with possible unknown

This work was supported by the European Commission under 601116-ECHORD++ (FlexSight experiment). The Authors are with the Department of Computer, Control, and Management Engineering “Antonio Ruberti“, Sapienza University of Rome, Italy. Email: lucamonorchio@gmail.com, danieleva2005@gmail.com, {imperoli, pretto}@diag.uniroma1.it.

[1] The classes are represented by a finite number of possible grasps.

objects, we aim to estimate the grasp that maximizes the probability to produce a seal between the gripper suction cup and one of the objects included in the scene. In order to acquire a suitable dataset, we exploit an automatic trial-and-error acquisition procedure in which an anthropomorphic robot performs a set of "quasi random" grasps: the vacuum sensor provides an answer on the state of the grasp (succeed or failed, i.e., the label of the example). Similarly to [11], we cast the grasp synthesis as a set of regression problems solved at once. A CNN, trained using *both* the grasping successes and failures examples, estimates with a single pass a set of grasps, each one related to a specific image tile. Along with the grasping point and the approach angles, for each image tile our method provides a confidence score that reflects how confident the model is that such configuration brings to a successful pick. We propose a loss function that enables the model to learn both what is an effective grasp and what is a surely ineffective grasp. We collected a first dataset composed by 4000 grasp attempts: our preliminary experiments show that, despite the reduced size of the dataset compared to the number of objects taken into consideration, our approach is able to synthesize grasps that lead to a 60% success rate.

A. Related Work

The approaches proposed to face the grasp synthesis problem can be divided into two categories: analytical methods and empirical methods. Being an empirical approach, in this work we revise only methods that belong to this category, with a focus on deep learning based methods. A comprehensive literature review about analytical methods can be found, e.g., in [14], while a survey about empirical methods presented in the pre-deep learning era can be found [2].

One of the first data-driven synthesis approaches based on learning and vision has been presented by Saxena *et al.* [15]. They proposed to employ a logistic regression model that uses visual features to predict from images good grasping points for a manipulator equipped with a parallel plate gripper. They synthetically generated training images along with ground truth grasps by using a computer graphics ray tracer. Jiang *et al.* [3] proposed to learn an oriented "grasping rectangle" for parallel plate grippers from RGB-D images using a Support Vector Machine as ranking algorithm, trained on a dataset of manually labeled images with correct and incorrect ground truth grasping configurations. The same dataset has been exploited in [5] and [11], two grasp synthesis approaches based on deep learning. Lenz *et al.* [5] exploit a two-step cascaded structure with two deep networks to rank the candidate grasps. Redmon and Angelova [11] cast the problem of grasp synthesis into a regression problem, dividing the input RGB-D image into a cell grid, and training the CNN for learning one predictor for each cell.

The creation of a manually labeled dataset is an extremely time consuming activity: several recent approaches try to overcome this issue by introducing self acquired and labeled datasets [10], [6], or more accurate synthetically generated datasets [4], [7]. Pinto and Gupta [10] created a large, self acquired dataset composed by 50K trial and error grasps, and formulated the problem of grasping with a parallel gripper as a CNN-based, 18-way binary classification problem over images patches. Levine *et al.* [6] scaled up the self-supervised learning concept by collecting a very large dataset composed by 800,000 grasp attempts performed by a cluster of similar robots over the course of two months. This dataset has been then exploited to train a CNN used to determine how likely a given motion is to produce a successful grasp.
Johns *et al.* [4] cast the grasp synthesis as a classification problem over depth images, where a CNN is used to predict the grasp score for a large set of poses of a parallel-jaw gripper. The training data is generated by using physics simulation and depth image simulation with 3D object meshes. Recently, Mahler *et al.* [7] introduced a very large synthetic dataset that includes 6.7 million "grasp images", i.e., small depth images representing parallel-jaw grasps, automatically labeled by using analytic metrics. A CNN (called Grasp Quality CNN) trained with this dataset is used to determine the most robust grasp over a set of candidate grasp images. This framework has been very recently extended to deal with vacuum grippers [8], one of the very few works, along with ours, that uses this type of gripper.

B. Contributions

Our method takes inspiration from the self-supervised approaches presented above, by implementing an automatic procedure for data acquisition and labeling that, differently from other previous work [10], [6], exploits a robotic system equipped with a vacuum gripper. Similarly to [11], we cast the grasp synthesis into a grid based regression problem but, differently from [11], a) we use a vacuum gripper: b) our model has been improved both in architecture design and error metric evaluation, by introducing a novel loss function inspired by [12] that takes into account both positive and negative samples during the training phase.

II. Grasp Synthesis Model

We aim to solve the grasp synthesis problem for arm manipulators equipped with vacuum grippers by employing a self-supervised, data-driven regression approach that enables to predict grasps for known and unknown objects. Similarly to [10] [6], our model is trained using a dataset acquired by means of a trial-and-error procedure. We employ a CNN that directly learns the grasping configuration from RGB-D images of the working area. Differently from [11], we exploit the full 4D information without getting rid of any RGB channel. Such network is trained in a end-to-end fashion, where the error between wrong and correct pickup points is directly optimized. Similarly to [11], we perform a regression on the entire image, in particular, we split the image into a cells grid in order to achieve the estimation of a grasping configuration and a confidence score for each cell. The

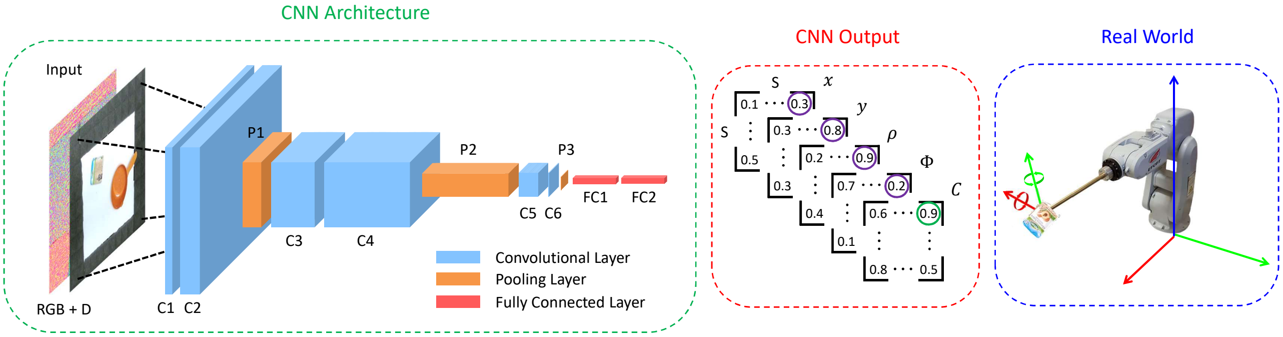

Fig. 2: The network is composed by 6 convolutional layers, 3 pooling layers and 2 fully connected layers in the very last part of the architecture. For the first part of the network, 2 max-pooling have been used, the last pooling layer instead performs an average of the previous feature map, so we have an avg-pooling before the input of the first fully connected layer. A ReLU activation function is used within all the convolutional layers while the fully connected ones use a sigmoid activation function. The network predicts a candidate point for each of the $S \times S$ cells in the image grid, and the one with the highest score (as depicted in the picture in green), or the highest four, will be taken as candidates for the final pickup (x, y, ρ, ϕ) end effector configuration.

confidence value reflects how confident the model is that such configuration brings to a successful grasp.

We introduces a new loss function inspired by the object detection method proposed in [12]: our loss minimizes the errors by considering positive and negative samples, giving the network the possibility to predict both successful and failing grasping positions by means of their relative confidence score.

A. Learning a Grasp Pose from RGB-D data

Given an RGB image $\mathcal{I}$ and a corresponding, registered depth map $\mathcal{D}$, we aim to estimate the grasp configuration that maximizes the probability to produce a seal between the vacuum gripper installed on a robot manipulator end effector and one of the objects included in the scene. Assuming a given hand-eye calibration, we can parametrize the grasp by means of the grasping position and orientation of the vacuum gripper (x, y, ρ, ϕ), and a confidence score C (see Fig. 2 for some details about the system work flow): x, y represent the image coordinates (in pixels) of the grasping point in the image reference frame, while ρ, ϕ represent respectively the gripper *roll* and *pitch* orientations relative to the grasping point in the robot reference frame, and C is the confidence score. The hand-eye calibration is used to map image coordinates x, y (in pixel) to world coordinates X, Y (in meters) in the robot reference frame; in this work, we look-up the third 3D component (i.e., the Z-component) of the grasping point directly from the depth image. The confidence score is defined in the range $[0, 1]$ and it tell us how confident the model is about the predicted point and orientation, how far is from an actual object in the scene, and also how accurate this prediction is for the network. We expect that where the score is close to 0, there should be no objects in such position or the grasp configuration will lead to a certain failure, conversely, when the score converges to 1, we expect that both the point belongs to an actual object and the orientation is suitable for an effective object grasp.

Each cell of the grid predicts a candidate grasping point and orientation. Although our first implementation of the system assumed that each prediction would fall into the limits of the cell itself, i.e. each cell just predicts a grasping point within its boundaries, we experienced not convincing results, due to the low accuracy of the network, far below the initial expectations. To overcome this problem, we allow that the predicted points and their relative roll and pitch orientation may also be inferred by theirs neighboring cells, namely each cell can now also predict points and orientations within the closest neighboring cells. This change allowed to significantly improve the accuracy of the network.

Layer	Input Size	Num. Filters	Filter Size	Stride
C1	424 × 424	16	7	1
C2	424 × 424	64	5	1
P1	424 × 424	-	2	2
C3	212 × 212	128	5	1
C4	212 × 212	256	5	1
P2	212 × 212	-	2	2
C5	106 × 106	64	5	1
C6	106 × 106	5	5	1
P3	106 × 106	-	4	4
FC1	1 × 1	320	-	-
FC2	1 × 1	320	-	-

TABLE I: The table shows the parameters of the layers in our model.

B. CNN Architecture

Our network is composed by 6 convolutional layers, 3 pooling layers and 2 fully connected layers in the very last part of the architecture (see Fig. 2 and Tab. I for further details on the network structure). For the first part of the network, 2 max-pooling have been used, the last pooling layer instead performs an average of the previous feature map, so we have an avg-pooling (average pooling) before the input of the first fully connected layer. A ReLU activation function is used within all the convolutional layers while the fully connected ones use a sigmoid activation function. We induce the network to predict a tensor with size $S \times S \times 5$, where S is the size of the grid (8 in our case). Given the presence of the sigmoid activation function in the last two dense layers, our model predicts only normalized values, in the range $[0, 1]$. This is consistent with the prediction of the confidence score, given that it should represents a probability value, but also with the prediction of the grasping point

(x, y) in the image and its relative gripper orientations (ρ, ϕ). Given a specific cell, if the x and y predicted coordinates are in the ranges $\{[0, 0.5), [0, 0.5)\}$ respectively, the network is actually predicting a point in the upper left neighboring cell, $\{[0.5, 1], [0.5, 1]\}$ for a point in the cell itself, and consequently for the top and right neighboring ones.

C. Grasping Loss Function

Our network architecture has been trained by minimizing both the position error on the x and y coordinates of the predicted point and the orientation error between the predicted roll and pitch end effector orientations w.r.t. the ground truth grasping configuration. We formulated the grasp synthesis as a regression problem, based on the following loss function:

$$\begin{aligned}
&\lambda_{coord} \sum_{i=0}^{S^2} \left[\left(x_i - \hat{x_i}\right)^2 + \left(y_i - \hat{y_i}\right)^2 \right] + \\
&\lambda_{orient} \sum_{i=0}^{S^2} \left[\left(\rho_i - \hat{\rho_i}\right)^2 + \left(\phi_i - \hat{\phi_i}\right)^2 \right] + \\
&\qquad K_{pos} \sum_{i=0}^{S^2} \mathbb{I}_i^{pos} \left(C_i - \hat{C_i} \right)^2 + \\
&\qquad K_{neg} \sum_{i=0}^{S^2} (1 - \mathbb{I}_i^{pos}) \left(C_i - \hat{C_i} \right)^2
\end{aligned} \tag{1}$$

Eq. (1) minimizes both the position error and the orientation error between the ground truth grasping position represented by the parameters $\hat{x_i}, \hat{y_i}, \hat{\phi_i}, \hat{\rho_i}$, and the ground truth confidence score $\hat{C_i}$, that is $\hat{C_i} = 1$ for successful grasps, $\hat{C_i} = 0$ otherwise. In particular, for each cell in the grid, we evaluate the errors both on the position and orientation part, and on the confidence scores, trying to minimize the loss both on *positive* and *negative* grasp in the ground truth. Here for *positive* and *negative* grasp we mean respectively the samples in the ground truth where the robot actually successfully performed an object grasp or not.

In Eq. 1 the first two summations refer respectively to the errors on the (x, y) coordinates and on the (ρ, ϕ) end effector orientations. The third and fourth lines instead represent the squared difference from the predicted confidence score C_i with respect to the ground truth confidence score $\hat{C_i}$: these two terms are mutually exclusive and we basically enable the right contribution by imposing the $\mathbb{I}_i^{pos}$ parameter equal to 0 when considering a negative pickup position and orientation in the ground truth and to 1 when considering a positive one.

Moreover, we weight the different contributions by using a set of gain parameters, namely λ_{coord} and λ_{orient} for the position and orientation errors respectively. In our experiments we set both equal to 1. In order to balance the ratio of positive and negative samples in the dataset we also normalized them by mean of the two gains K_{pos} and K_{neg}, and we practically set them equal to::

$$K_{pos} = 1 - \frac{\text{\#positive samples}}{\text{\#total samples}} \tag{2}$$

$$K_{neg} = 1 - K_{pos} \tag{3}$$

Parameter	**Value**
Optimizer	Adam
Momentum	0.9
Decay	0.0
Learning rate	0.0001
Batch size	16

TABLE II: The parameters used during the train phase.

III. Self-Supervised Dataset

Our experimental test bed consists of a customized robotic cell consisting of a lightweight robot arm, an RGB-D camera (a Microsoft Kinect v2 in our case), and an aluminum chassis that supports both the robot and the sensor.

A. Hand-Eye Calibration

Both the 3D camera sensor and the robotic manipulator have been calibrated in order to obtain their relative position w.r.t. each other. We performed such calibration using a custom calibration pattern mounted as background of our objects scene. The calibration has been performed in a single shot computation, meaning that at each cycle of both the acquisition and testing phases, we calculate the position of the board by detecting the markers on it and by using such position as reference frame for both the sensor and the robot manipulator. In this way we compensate at every cycle possible movements of the sensor or of the background board during the acquisition stage.

B. Data Acquisition Protocol

We acquired our own grasps dataset by implementing an automatic and self-supervised procedure. In particular, the robot performed a series of attempts in order to grasp objects within the scene. First of all, a ”quasi random“ point is chosen within a region of interest by exploiting the region proposal module of a CNN-based object detector [13]. We then perform a security check by analyzing the depth-map in that specific point in order to avoid any possible collision with the environment. The 2D image point is then transformed into the 3D robot reference frame by using the hand-eye calibration parameters and the Z component provided by the depth map, while generating random values for the roll and pitch orientations. The 3D position along with the generated orientations are sent to the robotic manipulator that, after the pick up attempt, automatically self checks if the grasp has been successful or not by querying the vacuum sensor provided with the gripper; the label (i.e., successful or failed grasp) is consequently associated to the current data item.
The acquisition procedure has involved 30 different objects, chosen from common household objects categories. We collected more than 4000 (positive and negative) grasp examples.

IV. Experiments

In this section, we present preliminary results of our system, tested on-line in a real-world scenario.

Test Set	*Successes / Tot. Attempts*	*Successful Attempt Rate*
Known Objects	61 / 100	0.61
Unknown Objects	34 / 100	0.34

TABLE III: The table shows results considering only the candidate with the highest score among all the predictions.

Finding the best grasping pose for unknown objects is not an easy task, moreover we want to estimate not only the position but also the orientation of the end effector, and it is something that intuitively cannot be directly inferred from the 2D information given by an RGB image. What we expect by testing our method is that the network would learn enough to discriminate between different objects' shapes, and for them just try to regress over the most similar point and orientations presented during the training phase for similar objects.

Following this intuition, we tested our network by presenting both known and unknown objects and collected performance results using two different metrics:

1) **The first best candidate:** the candidate tuple of predicted $\hat{x}, \hat{y}, \hat{\rho}, \hat{\phi}$ that has the highest confidence score is considered for attempting the grasping during the test phase;
2) **The four best candidates:** we consider all the first 4 candidates with the highest scores and try them during the test phase.

We implemented our model using the TensorFlow framework [1]; we trained the network described in Sec. II-B with the acquired dataset (see Sec. III), using the set of parameters listed in Tab. II.

The system has been tested on 100 different scenes that include randomly placed known objects (i.e., objects included in the training dataset) and on 100 different scenes with randomly placed unknown objects.

A. Experiments Results

Following the aforementioned metrics, results will be given in terms of grasping success rate. A single test is considered a successful grasp if, given the best grasping configuration (x, y, ρ, ϕ, C) predicted by the network, the robot manages to pick up an object in such position and end effector orientation lifting it up to a height of 30 cm over the board. As described in Sec. II-A, the Z component of the grasp is extracted directly from the depth map, since our current approach is not meant for learning and predicting also this component.

Tables III and IV show the results of our experiments. More in detail, in Tab. III we report the performance of our model by considering only the predicted candidate with the highest confidence score. The test have been performed both on the set of scenes with known objects, reporting a success rate of 61%, and on the set of scenes with unknown objects, reporting a success rate of 34%: these results, even if not impressive, can be considered interesting taking into account the reduced size of the used training dataset with respect to other datasets normally used in similar experiments [6], [10].
If we consider the second metric, and test the network by performing grasping attempts by using all the four predicted grasps with the highest confidence scores, the performance increases achieving a 21% and 22% improvement in the known and unknown objects cases, respectively. The fact that several of the best ranked grasp configurations enable successful grasps suggests that the model is learning the task in the correct way.

In Fig. 3 we report some qualitative results of some of the tests in order to show how the predictions are distributed. In particular, we can see how the model learned to correctly predict good candidates only in the area where objects actually are present. Moreover, we showed also to which cells each prediction relies, in *blue, green, yellow* and *orange* we plot the points with the *first, second, third* and *fourth* highest confidence score, respectively.

B. Discussion

Experiments denote how our model learns to predict positions and orientations suitable for the grasping of the objects, both in the case of known and unknown objects. The best results obviously come from objects that are known to the network, namely the ones it was trained with. On the other hand, the reduced size of the acquired dataset (4000 data items) could easily bring the network to overfit the input data: with this in mind, we believe that the behavior learned by the network has been in line with our expectations. Moreover, in some cases, scenarios that fail due to an incorrect output considering only the best candidate metrics, can be successful for some other candidates below the more confident one (see Tab. IV and Fig. 3). This observation opens the door to further consideration for improving the methods, starting from data augmentation, both concerning new acquisitions and synthetic generated approaches, coming to network design and loss function advances.

V. Conclusions

In this paper, we presented a method for solving the grasp synthesis problem for known and unknown objects by employing a self-supervised, data-driven approach. We collected training data by means of a trial-and-error procedure using a 6 d.o.f. robotic manipulator equipped with a vacuum gripper. We proposed to employ a CNN that directly learns the grasping function from RGB-D images by exploiting the image features as a regression problem, in particular we split the image into a cell grid to achieve the estimation of a grasping position for each cell of the grid. Depending on the predicted confidence score, the network is capable of producing both positive and negative grasping positions as output, namely the method has learned to predict both correct grasping positions and negative ones. We empirically verified that the method is actually capable of predict a correct grasp if considering the best 4 candidates metrics in the 82% and 66% of the

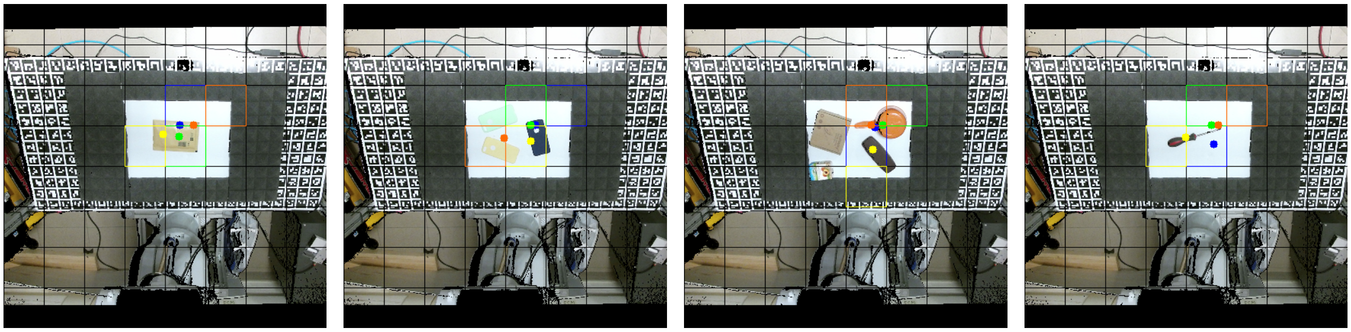

Fig. 3: The picture shows the prediction provided by the network, in particular here we depicted the candidates with the four highest confidence scores in different colors, with the following order (starting from the highest score): *blue, green, yellow* and *orange*. We colored with the same color also the cells responsible for these predictions.

	Known Objects Set		**Unknown Objects Set**	
Candidates	*Successes / Tot. Attempts*	*Successful Attempt Rate*	*Successes / Tot. Attempts*	*Successful Attempt Rate*
Only the First	61 / 100	0.61	34 / 100	0.34
The first 2	70 / 100	0.70	50 / 100	0.50
The first 3	78 / 100	0.78	62 / 100	0.62
The first 4	82 / 100	0.82	66 / 100	0.66

TABLE IV: The table shows results considering the 4 best candidates with the highest score among all the predictions. In particular, an attempt has been considered successful if among the n candidates the robot actually achieved in picking up the object from the scene.

tests for scenes that include known and unknown objects, respectively.

Further development can be done to this preliminary work. An extension of the dataset using a multiple robots setup will drastically reduce the data acquisition time. Moreover, data augmentation could provide us with further improvements in terms of saving time and resources, taking also into consideration the synthetic generation of crucial parameters such as vacuum gripper orientation. In addition, network behavior should be tested with a wider range of unknown objects, in hostile and therefore complicated scenarios such as industrial environments where many objects are messed up in a container and many occlusions and clutter occur.

References

[1] M. Abadi and *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[3] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. of IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011, pp. 3304–3311.

[4] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 4461–4468.

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[6] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. of Robotics: Science and Systems (RSS)*, 2017.

[8] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Proc. of ieee int. conf. on robotics & automation (icra)," in *Dex-Net 3.0: Computing Robust Vacuum Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning*, 2018.

[9] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[10] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proc. of IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2016, pp. 3406–3413.

[11] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. of IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[13] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[14] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326 – 336, 2012.

[15] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.