Transfer and Continual Supervised Learning for Robotic Grasping through Grasping Features

L. Monorchio[1] , **M. Capotondi**[2*] , **M. Corsanici**[2] , **W. Villa**[1] , **A. De Luca**[2] , **F. Puja**[1]

[1] Konica Minolta Laboratory Europe, Rome, Italy

[2] DIAG, Sapienza University of Rome, Italy

{capotondi, corsanici, deluca}@diag.uniroma1.it

{Luca.Monorchio, Wilson.Villa, Francesco.Puja}@konicaminolta.it

Abstract

We present a Transfer and Continual Learning method for robotic grasping tasks, based on small vision-depth (RGBD) datasets and realized through the use of Grasping Features. Given a network architecture composed by a CNN (Convolutional Neural Network) followed by a FCC (Fully Connected Cascade Neural Network), we exploit high-level features specific of the grasping tasks, as extracted by the convolutional network from RGBD images. These features are more descriptive of a grasping task than just visual ones, and thus more efficient for transfer learning purposes. Being datasets for visual grasping less common than those for image recognition, we also propose an efficient way to generate these data using only simple geometric structures. This reduces the computational burden of the FCC and allows to obtain a better performance with the same amount of data. Simulation results using the collaborative UR-10 robot and a jaw gripper are reported to show the quality of the proposed method.

1 Introduction

Automatic manipulation of objects is one of the main tasks that every robotic system has to accomplish in order to interact with the environment and cooperate with a human. As a generic definition, a grasping task requires a robot manipulator equipped with a gripper and sensors (e.g., a RGB camera and depth sensor) to pick up, move, and place down an object between assigned Cartesian poses in an unstructured environment (see Fig. 1).

This basic problem has been tackled first with analytical methods, mostly using vision and force feedback [Kragic and Daniilidis, 2008; Kuffner and Xiao, 2008]. However, the presence of multiple kinematic and dynamic constraints has limited so far their success, restricting the field of application to particular types of objects in well-defined environments. More recen1tly, following the widespread diffusion of data driven methods for image processing, the problem has been addressed through Deep Learning (DL) techniques,

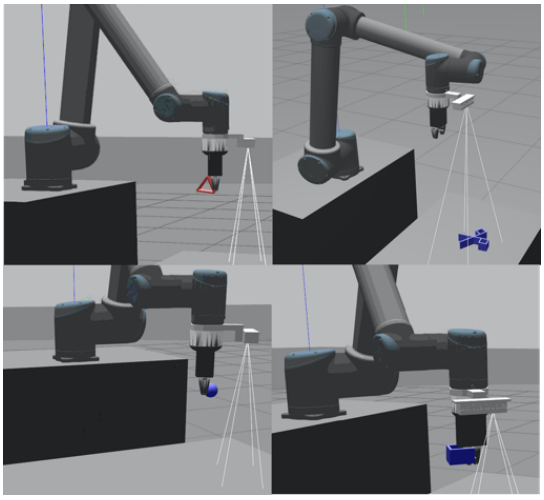

Figure 1: Snapshots of the simulated UR-10 robot equipped with a jaw gripper performing the grasping procedure on different objects.

as in [Bohg *et al.*, 2014]. Within this paradigm, Supervised and Unsupervised Learning and also Reinforcement Learning were used.

Both Supervised and Unsupervised methods are based on classification and regression of the grasping pose, usually parametrized by the Tool Center Point and the approach vector, as assigned to a set of successful previous grasping experiences. In supervised methods, the datasets are generated within a procedure that is independent from the learning process, whereas in unsupervised methods, the same learning algorithm is queried in inference for generating the grasping policy and for producing samples accordingly. On the other hand, Reinforcement Learning realizes the search of a policy for generating either the grasping pose or the entire movement through a mechanism of trials and rewards [Quillen *et al.*, 2018].

In order to improve efficiency and to limit data hungriness of the above learning techniques when applied to several different objects in industrial and human environments, two special DL concepts have been developed for generalization and knowledge storage: Transfer Learning (TL), which consists in the transfer of knowledge from a certain domain or task to another, and Continual Learning (CL), which ad-

*Contact Author

dresses the storage of previous knowledge after changing the application task [Pan and Yang, 2010; Delange *et al.*, 2021 to appear]. Both methods were found to be very useful when implementing learning algorithms in real-world applications, limiting the need of retraining and improving the reuse of acquired knowledge. This is particularly relevant when the data are generated by means of physical trials of robotic manipulation, a procedure that is time consuming and potentially dangerous.

In this paper, we present a method to realize Transfer and Continual Supervised Learning in the context of vision-depth based robotic grasping tasks. Different CL methodologies have been exploited for robotics, as surveyed by [Lesort *et al.*, 2020]. In our work, we focus on regularization-based techniques. In particular, a predefined architecture composed by a Convolutional Neural Network (CNN) followed by a Fully Connected Network (FCC) is used to process RGBD images, and its main properties are exploited so as to become as little data hungry as possible and, at the same time, still completely aware of the task. Differently from Yen-Chen *et al.* [2020], we propose here to modify the classical transfer learning procedure by using high-level features that we call Grasping Features (GFs). These are generated from the CNN during the training on a specific grasping task, instead of the visual features generated by training on datasets suited for image recognition and segmentation [Saxena *et al.*, 2008].

The use of Grasping Features introduces many advantages both for Transfer and Continual Learning. In fact, instead of completely decoupling the grasping task from its features, delegating the CNN to visual recognition only and thus forcing the FCC to learn both the grasping task and the variability of the objects to be grasped, the GFs already encode information about the grasping task, allowing the FCC to be fine tuned only for coping with object variety. As a result, this allows the use of smaller sets of data. On the other hand, the main limitation to the use of GFs in TL is that datasets composed by samples of robotic grasping are very uncommon and also quite dependent on the dataset generation policy —as far as we know, the only open dataset for grasping is the one provided by the Cornell University [Jiang *et al.*, 2011].

Because of the restricted availability of data and in order to obtain an efficient implementation of the proposed method, we introduce a strategy for generating larger datasets that uses only grasping of basic objects with simple geometry in a procedure that we call Shape Decomposition. Our claim is that the use of Grasping Features and the generation of a knowledge base through Shape Decomposition allow an efficient pipeline of Continual Learning in cascade to Transfer Learning even with relatively few data. In fact, the use of relevant high-level features in a more informative dataset reduces the training effort of the FCC. Fine tuning on new objects will become easier, with improvement in TL performance, while saturation of the FCC weights will occur later, offering thus more freedom to regularization and preventing catastrophic forgetting events.

The paper is organized as follows. In Sec. 2, we formulate the grasping problem and introduce the basic assumptions of the solution method. In Sec. 3, the learning network architecture is defined, focusing in particular on the dataset generation procedure and on TL and CL issues. The implementation details are described in Sec. 4, while Sec. 5 reports the results obtained in a simulated environment. Conclusions and future work are summarized in Sec. 6.

2 Problem Formulation

We introduce here the formulation of the grasping problem and the main working assumptions underlying our solution method.

While grasping is defined in many ways in the literature, for the sake of this paper we consider the following definition. A grasping task consists in the use of a robotic system to pick up and move a certain class of objects placed on a flat surface, using a gripper and an RGBD camera mounted on the robot end effector. Such definition is technically independent from the type of robot used for the task, depending only on the gripper structure, the grasping policy, the object properties and the camera features.

Real objects have usually very complex shapes with different mechanical properties (such as weight or CoM position), so that the grasping policy cannot be the same for all objects and a specific algorithm for its generation is required. We define the grasping policy as follows. Consider a robot end effector mounting a gripper and carrying a RGBD camera that provides RGB and depth images. The grasping policy consists in an algorithm that provides a Grasping Tuple (GT) $[x, y, \theta]$ corresponding to the desired Cartesian coordinates and orientation of the gripper in the image plane. According to the depth measurements and image coordinates, the GT is transformed into the world frame of the robot. A planner computes a feasible path for the robot end effector, with resulting commands issued in the joint space. The robot will execute the motion from its home position to the desired end-effector pose and will try then to grasp the object, according to the particular structure of the gripper (e.g., with jaws, vacuum suction, or multiple fingers).

With this definition of a grasping task, the problem is now shifted to the regression of the functional map that generates suitable values of the GT. In the following, we will introduce improvements that are valid in principle for any data driven algorithm used to solve the GT regression problem, assuming that RGBD images are processed by a neural network with a particular structure.

3 Network Architecture

We specify here the type of architecture we have considered in our experiments, together with the method of Transfer and Continual Learning we exploit.

With reference to Fig. 2, the network architecture for solving the GT regression problem is composed by a Convolutional Neural Network (CNN), that takes as input the stack of RGB and depth images provided by the camera, and by a Fully Connected Cascade Neural Network (FCC), that takes as input the output of the CNN, returning the predicted Grasping Tuple as output.

It is well known that CNNs are able to pre-process efficiently images, extracting high-level features of images such as vertices, angles, shapes, or color Goodfellow *et al.* [2016].

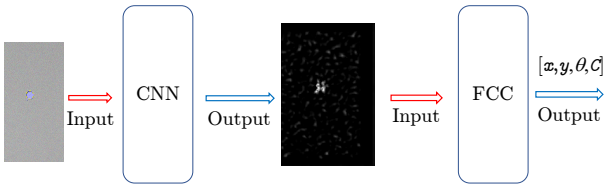

Figure 2: Scheme of a generic network architecture, in which image pre-processing is executed by the CNN and fed as input to the FCC that returns as the Grasping Tuple (including the confidence level C) as output.

In our architecture, the CNN goal is still to encode the high-dimensional image input to a low-dimensional output for the cascaded processing by the FCC, but we let the CNN produce an intermediate output that eases the FCC task of generating the desired grasp.

For this, we use as starting point the network described in Monorchio *et al.* [2018], on top which we build the entire TL-CL pipeline. This architecture is structurally similar to the YOLO introduced by Redmon *et al.* [2016], but with a different goal. It takes as input RGBD images and returns as output a series of Grasping Tuples (GTs) stacked with their confidence level, a value that quantifies how much the network is confident about the returned prediction. The input images are divided in cells, each associated to a specific prediction and centered in different image coordinates. The architecture returns as output a tuple composed by the relative position of the gripper (x, y) with respect to the cell, its absolute orientation θ, and the associated confidence C. The Grasping Tuple is thus redefined as $[x, y, \theta, C]$ and the addition of confidence values in the output modifies slightly the regression. For the robot grasping, the relative positions predicted by the network are transformed in world positions according to the RGBD measurements.

While in YOLO proposals are pruned according to the predicted probability of the single bounding box, in our structure all predictions are considered individually and sorted by increasing confidence. Keeping more proposals is not a drawback in general, since in many industrial cases multiple grasps are acceptable (or convenient) –a typical example is the task of emptying bins from a container.

The supervised learning procedure minimizes a loss function L_{task} which is the weighted sum of quadratic terms, one for each component of the tuple, corresponding to the squared difference between network prediction and ground truth:

$$\begin{aligned} L_{\text{task}} = \lambda_{\text{position}} \sum_{i=0}^{S^2} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)] \\ + \lambda_{\text{orientation}} \sum_{i=0}^{S^2} (\theta_i - \hat{\theta}_i)^2 + \lambda_{\text{confidence}} \sum_{i=0}^{S^2} (C_i - \hat{C}_i)^2. \end{aligned} \tag{1}$$

In (1), the sum is over the set of all predictors (i.e., all cells) in the image, $[\hat{x}_i, \hat{y}_i, \hat{\theta}_i, \hat{C}_i]$ corresponds to i-th ground truth values, and the (positive) hyper-parameters λ_{coords}, $\lambda_{\text{orientation}}$, and $\lambda_{\text{confidence}}$ balance components (and units) in the loss function.

3.1 Dataset generation

In order to correctly fit the GT policy, we followed a (self) supervised learning approach, automatically generating the dataset in a decoupled way with respect to the training procedure. Consider an input RGBD image specified by

$$X \in R^4 \times R^{\text{P}},$$

with the R, G, B, and D data of a camera having a total of P pixels, and a tuple

$$Y = [x, y, \theta, C],$$

where x and y correspond to the Cartesian coordinates of the grasping position in the image reference, θ is the roll angle of the gripper, and C is the confidence value, a real number in $[0, 1]$. The regression problem corresponds to approximating the map

$$f : X \to Y.$$

In order to collect samples of this map, a quasi-random policy is employed, with a bounding box generated around the object using selective search algorithms [Uijlings *et al.*, 2013] and grasping positions randomly picked inside the box. As for the gripper orientation, a random value is drawn from the interval $(-\pi, \pi]$. For each proposed GT, a grasping trial is executed, associating $C = 1$ in case of successful grasping and $C = 0$ otherwise. All (positive and negative) trials are stacked in the dataset used for network training.

3.2 Supervised transfer and continual learning

For effective human-robot interaction and in industrial applications, scalability of the approach with respect to the increasing cardinality of the set of objects considered in the grasping task is an appealing property. Techniques for Transfer and Continual Learning have been exploited in order to avoid redundant training and allow the largest possible reuse of previous knowledge.

In particular, in our TL procedure for grasping, we trained first the entire network on a large set of objects having different shapes and structural properties (Baseline Training). Afterwards, in order to realize a fine tuning on unknown new objects, we trained again the FCC while keeping the previously obtained weights of the CNN fixed. The main idea of this approach is similar to what was realized for pure visual tasks [Huh *et al.*, 2016]. In the context of grasping, this approach has been exploited by Yen-Chen *et al.* [2020].

Moreover, among the many existing CL approaches [Lesort *et al.*, 2020], we resorted to a regularization of the FCC weights, according to our idea that the two different components of the network, CNN and FCC, are delegated to TL and CL, respectively.

In this context, the goal of CL is to solve multiple learning tasks in such way that the overall network, after training on new objects, preserves still an acceptable level of performance on previous grasping tasks. Regularization is achieved

by introducing soft constraints in the optimization, augmenting the loss function in (1) with a weighted distance to previous tasks parameters:

$$L = L_{\text{task}} + \lambda_{\text{reg}} \sum_{j}^{\text{previous tasks}} (\mathbf{p} - \mathbf{p}_j)^T \mathbf{W}_j (\mathbf{p} - \mathbf{p}_j) . \quad (2)$$

In (2), vectors $\mathbf{p}$ and $\mathbf{p}_j$ are, respectively, the actual and the j-th task weights of the FCC, $\mathbf{W}_j > 0$ is a weight matrix associated to the distance from the j-th task, and the hyper-parameter $\lambda_{\text{reg}} > 0$ balances the regularization term in the total loss.

4 Implementation

We provide here some details on the proposed method, in particular how grasping features differ from visual ones and the rationale of Shape Decomposition.

4.1 Grasping features

One of the main properties of CNN image pre-processing is the partial explainability of the obtained features after the network filtering and elaboration [Zhang *et al.*, 2018]. Visualization of the network outputs can be very expressive about how the algorithm analyzes and processes images [Zeiler and Fergus, 2013]. These concepts are very important for the heuristic justification of transfer learning approaches. As shown by Razavian *et al.* [2014], features extracted by a CNN trained on large datasets (such as ImageNet) improve generalization, allowing network adaptation to new objects by retraining only the FCC weights.

In contrast to image processing (such as segmentation or classification) intended only for vision goals, we have added the use of Grasping Features resulting in a different output of the CNN. We illustrate this explicitly in Figs. 3 and 4. A visual comparison is shown between the features extracted by the CNN of a YOLO network, trained only for image recognition and classification, and the GFs extracted by the modified architecture proposed in Sec. 3.

In both the considered situations, namely with a set of real objects and with a single simulated object with noise, the features extracted by the two approaches are quite different. In fact, while the output of the CNN activation functions for image recognition just extract object shapes, the GFs obtained by the CNN trained for grasping show peaks at image points which are more suitable for the task.

Figure 3: From left to right: Input image of a set of real objects, the visual features extracted from the CNN of a YOLO network, and the associated Grasping Features. The GFs have intensity peaks at the image points of possible grasp positions, while visual features recover just object shapes.

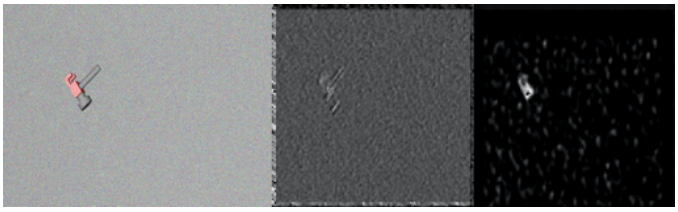

Figure 4: From left to right: Noisy input image of a simulated hammer, the visual features extracted from the CNN of a YOLO network, and the associated Grasping Features. In the presence of a single object, the GFs have even more distinctive peaks at the image points corresponding to possible grasp positions (in this case the upper part of the hammer). The multiple smaller peaks are due to unfiltered noise.

Since the Grasping Features embed not only geometric information but also physical properties (such as mass, CoM position and inertia) of the object, they are more informative for the goal of the algorithm. Having a richer input available for the FCC, the same amount of data will induce a faster convergence of its fine tuning. As an expected result, transfer learning is executed with fewer samples (in the order of thousands, instead of millions).

4.2 Shape decomposition

The use of large datasets of images allows to decouple image pre-processing from grasping policy learning, as realized, e.g., in [Yen-Chen *et al.*, 2020] where all the effort is delegated to the subsequent FCC. Despite the better adherence of Grasping Features to the task allows to distribute the computational burden between CNN and FCC, the main limitation of their application is the scarce availability of grasping datasets. As a matter of fact, GFs reduce but do not eliminate the need of large sets of samples for transfer learning.

We propose to face this issue by generating grasping samples in a convenient mode. Usually, objects encountered in real environments may be decomposed into simpler geometrical shapes, such as spheres, cylinders, or boxes, some of which dominates over the others from the point of view of size and/or physical properties. Figure 5 illustrates visually such concept.

The use of these basic objects improves the efficiency of the dataset. In fact, it is more convenient to approximate the grasping of complex objects as nonlinear combinations (via the FCC) of grasping features extracted from simpler objects rather than vice versa. This is particularly evident in the case of small datasets.

Instead of extracting the correct grasping features from general images using the strength of a large dataset, we decided to use a smaller set of samples but with basic shapes that are similar to the features we expect to extract from the network. Such use of a reduced dataset appears opposite to the general trend in deep learning image-based approaches. Nonetheless, this turned out to be a practical and successful procedure both for TL and CL. In particular, it allows to realize an efficient Continual Learning through regularization methods.

Indeed, random high-level features fed in input to a large

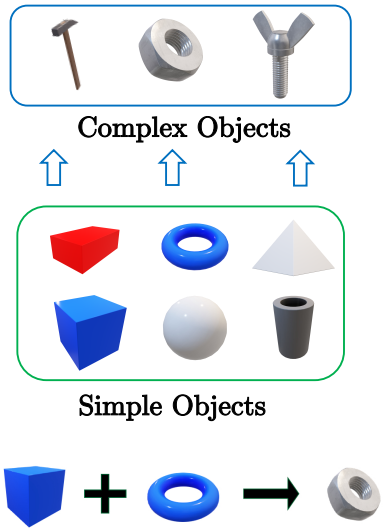

Figure 5: Shape Decomposition: Simple objects may be combined to generate complex objects, with an example of a bolt obtained as the overlap of simpler shapes.

FCC network can still be combined in order to approximate the grasping policy for any single object. However, when learning over time the grasp for multiple objects, the regularization process would become almost useless. Because of the low significance of the provided features, the positions in the parameter space of the individual minima of the loss function 1 and of the regularization component in 2 are in general very separated, thus leading to a spurious solution of the optimization problem. The results in the next section support empirically the validity of our approach.

5 Results

We present here results of numerical tests obtained with a Gazebo simulator, using a 6-dof UR-10 robot equipped with a jaw gripper.

5.1 Transfer and continual learning pipeline

The test pipeline with the cascade of transfer and continual learning is shown in Fig. 6. The test procedure is as follows. First, we use a previously generated baseline dataset for training the entire network (CNN and FCC) so as to extract Grasping Features. Then, the weights of the convolutional layers are frozen and the FCC is trained sequentially on new objects, applying on each new training a regularization with respect to previous items —see eq. (2). The classification of objects as new or old is realized manually by the user, and datasets are provided accordingly.

For this, we used very small datasets, around 1000 samples for each object (a very small figure when compared to classical transfer learning and fine tuning approaches).

The final system has been tested on a simulated scenario containing all the objects previously encountered. The robot tries to grasp each of the objects and performance is evaluated according to the relative frequency of positive results.

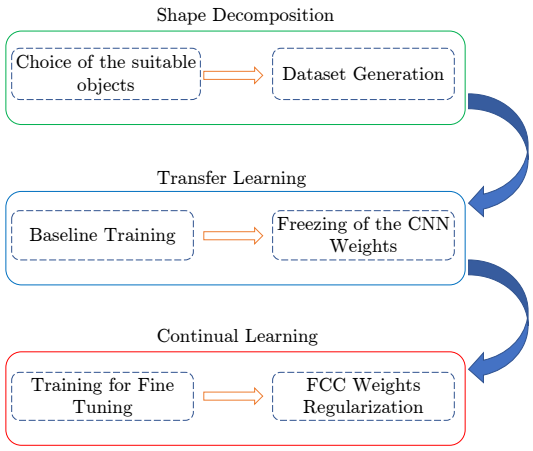

Figure 6: Pipeline to evaluate the proposed approach for realizing Transfer and Continual Learning.

5.2 Simulated tests

The tests were realized using a set of 11 objects, divided in simple objects (bar, sphere, triangle, cylinder, box, T-shape) and complex objects (hammer, modified T-shape, modified box, cup, scissor), see also Fig. 7. For grasping, the robot uses a very common parallel gripper with two jaws to hold a workpiece.

The Continual Learning procedure is implemented according to Kirkpatrick *et al.* [2017], as modified by Huszár [2018]: regularization is performed using only the weights of the last task, the regularization weight is itself a weighted sum of the Fisher matrices of all previous tasks. As for the training algorithm, we used Adam optimizer with a starting learning rate of 0.001, parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a batch size of 4 images, 15 epoches for the baseline training, and 7 epoches for the cascade of regularized training.

We define the score of the algorithm for each object as the frequency of times (normalized to 1) that the robot successfully grasps the object. Since our architecture generates N predictions for the N cells in the image, we considered in

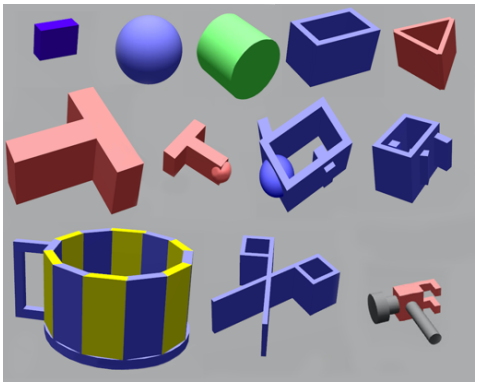

Figure 7: Different views and scales of the objects used for the tests. The first row shows simple objects, while complex/combined objects are shown in increasing complexity in the second and third rows.

each trial the first 4 grasp proposals with the highest confidence levels. Accordingly, a positive outcome is recorded when at least one of these grasps is successful.

We compared two learning strategies, see Fig. 8. In the first one, we used complex objects for the baseline training and then regularized training on the simple objects, in the same order as they have been listed above. The results in the top histogram of Fig. 8 show that the performance is acceptable despite the few training data, except for some degradation on certain objects (in particular, the bar and the cylinder). Each column in the histogram includes the outcome using partial ordered subsets or the totality of the 4 grasp proposals. In the second strategy, we switched to simple objects for baseline training and complex ones for the regularization, again in the same order as before. The performance is globally improved (center histogram in Fig. 8), and the degradation for the same previous objects is reduced. The last histogram compares the scores of the two strategies, considering for each object all 4 grasp proposals in the trials.

Summarizing, both strategies give a merit to the benefit of using Grasping Features in order to reduce the size of the training dataset. Moreover, the final score is improved when the baseline dataset is composed by simple objects, according to the Shape Decomposition idea, followed by the use of complex objects in the CL pipeline.

6 Conclusions

We have presented an original method for realizing Transfer Learning (TL) and Continual Learning (CL) for RGBD-based robotic grasping tasks. The basic idea is to define a more efficient transfer learning not using just visual features extracted from a common image databases (such as ImageNet) but instead with grasp-oriented datasets that embed characteristic Grasping Features extracted for the specific tasks.

In order to overcome the reduced availability of this type of grasping data, we introduced also the idea of generating samples using only simple geometric shapes through a Shape Decomposition procedure. This is somewhat opposite to the current trend in deep learning approaches: we avoid the critical (time-consuming and/or hazardous) generation of huge datasets using a physical robot by providing more informative reduced sets of samples.

Our obtained results show that TL is possible in this way also with very few data (thousands instead of millions). Moreover, by comparing how much the TL+CL pipeline is affected by the baseline dataset, we have shown that Shape Decomposition positively affects performance.

Though promising, these results are indeed preliminary and many improvements are possible. In the next future, we plan to use a more realistic simulator for improving generalization properties and coping with the well-known problem of SimToReal. Moreover, we will analyse the proposed approach for larger classes of objects (in the order of hundreds). Finally, we would like to use other CL methods, to highlight how much the Grasping Features and the Shape Decomposition are adaptable to different algorithms.

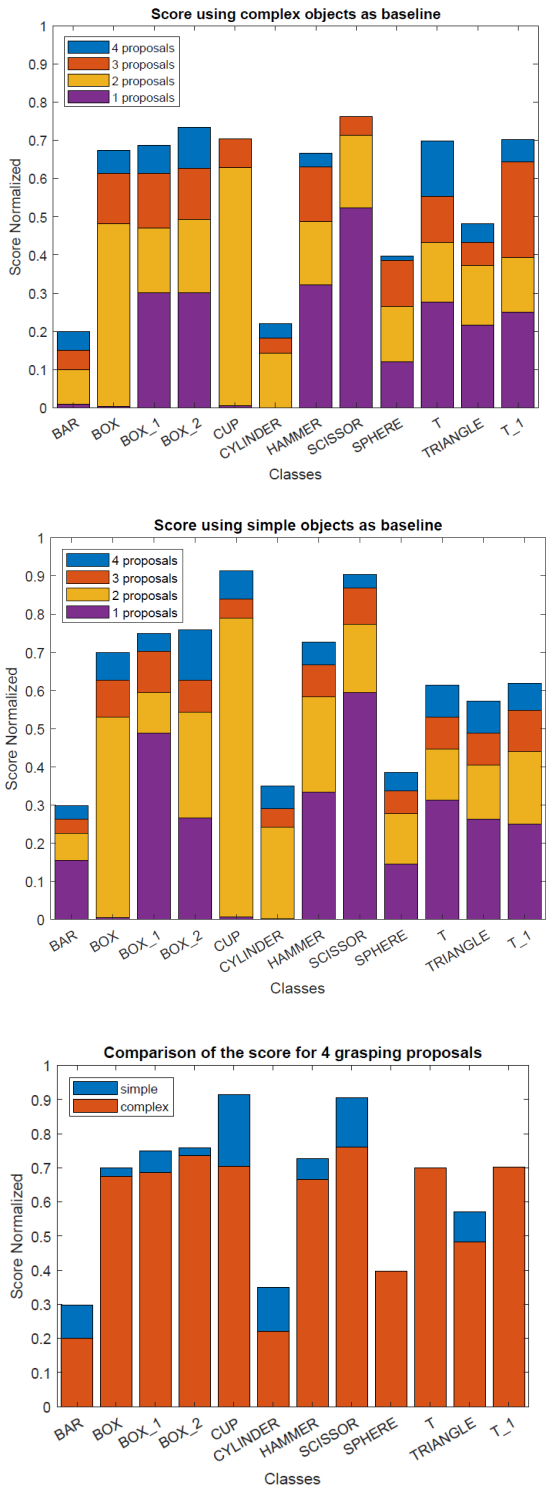

Figure 8: Performance histograms for different baseline datasets: Using complex objects (top) and using simple objects (center). Color codes are used to specify the number of grasp proposals (GTs) used in each trial. The bottom histogram compares the scores in the two cases comparison using all 4 grasp proposals in the trials.

References

Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (to appear).

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Mi-Young Huh, Pulkit Agrawal, and Alexei A. Efros. What makes ImageNet good for transfer learning? *CoRR*, abs/1608.08614, 2016.

Ferenc Huszár. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, 115(11):E2496–E2497, 2018.

Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International Conference on Robotics and Automation*, pages 3304–3311, 2011.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Danica Kragic and Kostas Daniilidis. 3-D vision for navigation and grasping. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 812–822. Springer, 2008.

James Kuffner and Jing Xiao. Motion for manipulation tasks. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 898–923. Springer, 2008.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020.

Luca Monorchio, Daniele Evangelista, Marco Imperoli, and Alberto Pretto. Learning from successes and failures to grasp objects with a vacuum gripper. In *IEEE/RSJ IROS Workshop on Task-Informed Grasping for Rigid and Deformable Object Manipulation*, 2018.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation*, pages 6284–6291, 2018.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519, 2014.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

Uijlings, J.R.R., van de Sande, K.E.A., Gevers, and T. et al. Selective search for object recognition. *The International Journal of Computer Vision*, 104(2):154–171, 2013.

Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation*, pages 7286–7293, 2020.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.