Bin Picking of Reflective Steel Parts using a Dual-Resolution Convolutional Neural Network Trained in a Simulated Environment

Jonatan S. Dyrstad[1,2], Marianne Bakken[3], Esten I. Grøtli[3], Helene Schulerud[3] and John Reidar Mathiassen[1,*]

***Abstract*— We consider the case of robotic bin picking of reflective steel parts, using a structured light 3D camera as a depth imaging device. In this paper, we present a new method for bin picking, based on a dual-resolution convolutional neural network trained entirely in a simulated environment. The dual-resolution network consists of a high resolution focus network to compute the grasp and a low resolution context network to avoid local collisions.The reflectivity of the steel parts result in depth images that have a lot of missing data. To take this into account, training of the neural net is done by domain randomization on a large set of synthetic depth images that simulate the missing data problems of the real depth images. We demonstrate both in simulation and in a real-world test that our method can perform bin picking of reflective steel parts.**

I. INTRODUCTION

Bin picking is the problem of grasping objects randomly placed in a bin. This is a problem that often occurs in industrial settings where objects come out of a production line packaged in bulk, without isolating individual objects, and where the objects are transported to a second production line that subsequently must isolate and process these objects individually. Due to the importance and relevance of the problem, bin picking has been well studied [19], [24]–[26] in the literature. Challenges in bin picking arise when seeking to develop a bin picking algorithm that can be automatically customized for specific objects, and when these objects are very reflective. We present a method for bin picking that addresses these two challenges.

The input to the grasp detection network is a depth image and the output is a set of possible 3D grasps (e.g. 5-DOF or 6-DOF gripper poses). The use of a dual-resolution network enables both high accuracy in a focus region of interest for placing the grasp and estimating the grasp pose, as well as enabling a low-resolution context awareness that e.g. ensures that the grasps do not collide with other objects in cluttered scenes. Fig. 1 shows the robot and the Zivid[1] 3D camera used in our experiment and the steel parts in our bin picking case.

We first evaluate our approach on simulated test data and then demonstrate it in an exemplary real-world scenario involving bin picking of steel parts using a robot with 5-DOF placement of a vacuum suction gripper. Training of the neural network is done entirely on synthetic depth images generated by domain randomization in a simulated environment. This

*Corresponding author, `John.Reidar.Mathiassen@sintef.no`
[1]SINTEF Ocean AS, Trondheim, Norway
[2]NTNU, Department of Engineering Cybernetics, Trondheim, Norway
[3]SINTEF Digital, Oslo, Norway
[1]http://www.zividlabs.com/

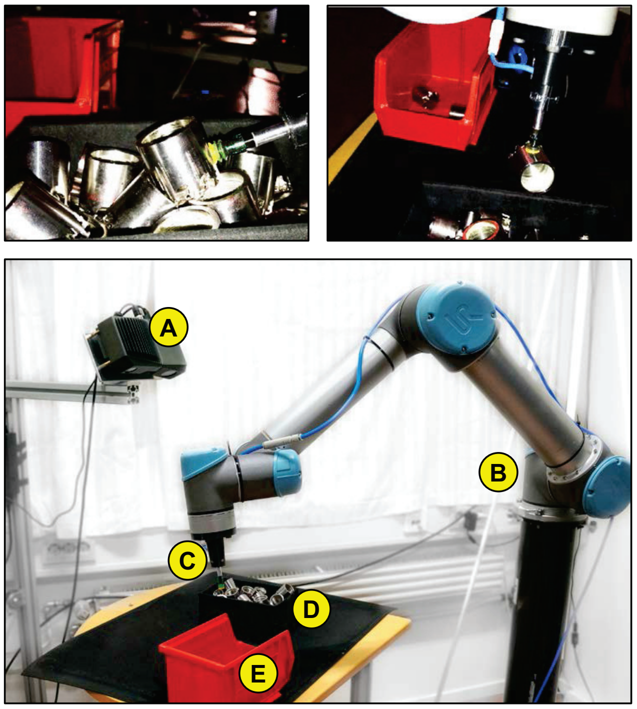

Fig. 1. Grasping steel parts with a suction gripper (top two images). An overview of the bin picking setup, including a Zivid 3D camera (A), a UR5 robot (B), a pneumatic suction gripper (C), a bin of reflective steel parts (D) and a bin (E) for placing the steel parts after picking.

approach is used to generate simulated data for training of the neural network [23] that will work well in the real world.

Our main contributions are:

- A dual-resolution convolutional neural network for end-to-end 5-DOF grasp estimation from depth images, which uses a high resolution focus network to compute the grasp and a low resolution context network to avoid local collisions.
- A simulation environment using domain randomization to automatically generate large data sets for training the neural network, given known reflectivity and geometric properties of objects in the bin-picking scenario.
- Demonstrating that the dual-resolution neural network can be trained entirely in a simulated environment on specific objects, and be deployed in a robot that performs bin picking of these objects in the real world.

Although our experiments are done using a suction gripper on smooth-surfaced metal objects, the methodology of our contributions should be applicable also for other types of objects and grippers - as long as these can be simulated. The rest of the paper is organized as follows: We discuss related work in Section II. We present our grasping method in Section III. We then describe our experimental setup and results in Section IV. The conclusion and suggestions for future work are in Section V.

II. RELATED WORK

Detecting robot grasps from 3D or depth images is an active research field, both in terms of using geometry-based methods [8]–[12], [14] and deep learning [1]–[5], [7], [13], [20], [21]. Geometry-based methods attempt to match 3D CAD models to point clouds to compute the object pose [16]. Some research suggests that primates and humans have separate neural pathways for object recognition and grasping [17], and the object detection and pose estimation has often been treated as an isolated problem separated from grasp selection in the bin-picking literature. Geometry-based methods have been well explored for pose estimation in bin-picking, such as Abbeloos et al [24], that uses the popular point pair feature approach, first presented by Drost et al. [27]. Buchhilz et al. [26] suggests a two-stage approach where the full object pose is estimated after grasping based on inertial features. On the other hand, Ellekilde et al. [25] focuses on the grasp selection alone and proposes a learning framework to improve on this part. A different approach is to detect a valid grasp directly from 2D or 3D images without explicit pose estimation. Domae et al. [19] estimates the graspability of an object based on depth maps without the assumption of a 3D model, which makes it applicable to all objects. Saxena et al. [6] developed a grasp detection algorithm based on extracted hand-coded features from stereoscopic cameras, and machine learning (logistic regression). Other hand-coded feature-based approaches using machine learning have also been developed [11].

Instead of hand-coding features, one may use deep learning to extract the relevant features for grasping [3]–[5], [7]. These works use deep learning on depth images and output grasping rectangles with center points parameterized by (x, y) and θ in the plane of the depth image, and use the depth image values within the rectangle to compute the distance to that point. For a parallel-plate gripper approaching perpendicular to the viewing plane of the depth sensor, this approach works well. In general, this may not necessarily work, and a full 3D grasp may be required. This has previously been solved by using deep learning [1]. Here the 6-DOF grasps are generated randomly within a volume of interest and a convolutional neural network is used to evaluate the grasps by inputting multiple projections of a 3D point cloud volume centered at the grasp. Levine et al. [22] uses a convolutional neural network to learn hand-eye coordination from a large dataset of grasp attempts with real robots and a large variation of domestic objects in semi-cluttered bins. In contrast to Pinto et al. [5] they use the trained network to servo the gripper in real-time, which makes it more robust to mistakes and moving objects. Other approaches [2], [21] also use deep learning to evaluate the quality of a grasp. This differs from our approach, in which we use deep learning to compute the 3D grasp itself. In terms of input-output domain of the neural network, the work most related to ours is Huang et al. [13], where the output is the robot hand position, rotation axis and angle of rotation. Our approach differs, in that we use a dual-resolution convolutional neural network. Dual-resolution networks have successfully been used to recognize hand gestures from depth images [15]. Relative to this, our deep learning approach is novel in that we integrate two resolutions into fully connected layers before computing the output, whereas [15] integrates the final output of each resolution.

The use of a dual-resolution network enables both high accuracy in a focus region of interest, when placing the grasp and estimating the grasp pose, as well as enabling a low-resolution context awareness that e.g. ensures that the grasps do not collide with other objects in cluttered scenes. This approach is in principle similar to the fast filter-based bin-picking algorithm in [19], which uses binary and linear contact and collision filters that consider the geometry of the gripper, to filter the depth images and thereby locate 4-DOF grasps. The principle differences between our work and [19], is that we use a neural network to provide end-to-end training of a 5-DOF (extendable to 6-DOF) grasp detection network that automatically designs the appropriate general and nonlinear filters relevant to grasp estimation and collision, in a way that considers both the geometry of the gripper as well as the geometry and reflectivity of the objects.

One challenge with deep learning for robot grasping is the lack of large datasets of labeled training data, especially for depth images. Pinto et al. [5] collected grasp attempts on a real robot to train their network. Schwarz et al. [28] take another approach, and uses pretrained models from object classification to output bounding boxes and object boundaries for further grasp detection. Our solution to this problem is to generate our own labeled depth dataset in a simulated environment. The benefit of this approach is that it enables us to do end-to-end learning of the grasp itself, and utilize the depth data more directly such that we can handle challenging surfaces.

Generation of realistic looking synthetic images is an active research topic and [29] has proposed a method based on an adversarial network to improve the realism of simulated images using unlabeled real data.

Compared to the literature (e.g. [21]) our network does not require singulating or segmentation of the objects before estimating the grasp. Segmentation is difficult when the objects are highly reflective, since one is not guaranteed to have contiguous depth measurements within an object due to missing data in the depth images. To solve this problem, our neural network is trained entirely by domain randomization in a simulated environment that explicitly simulates the reflectivity of the objects in cluttered scenes by rendering missing data in the depth images similarly to how missing data occurs in real 3D images of reflective objects.

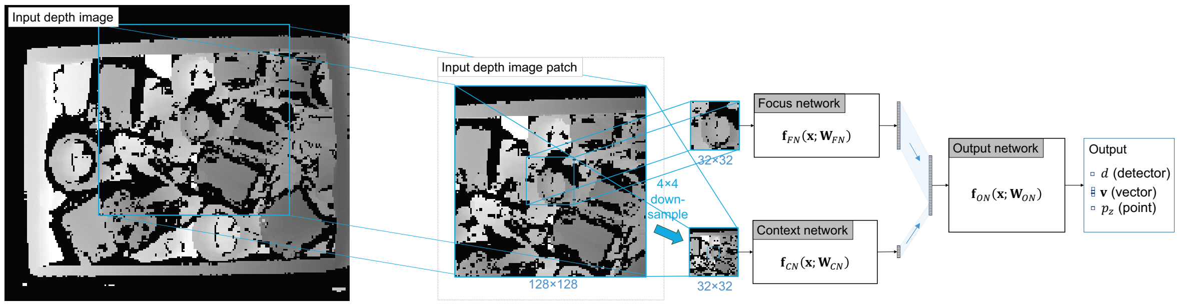

Fig. 2. Overview of the dual-resolution neural network, including scaling of input image patches.

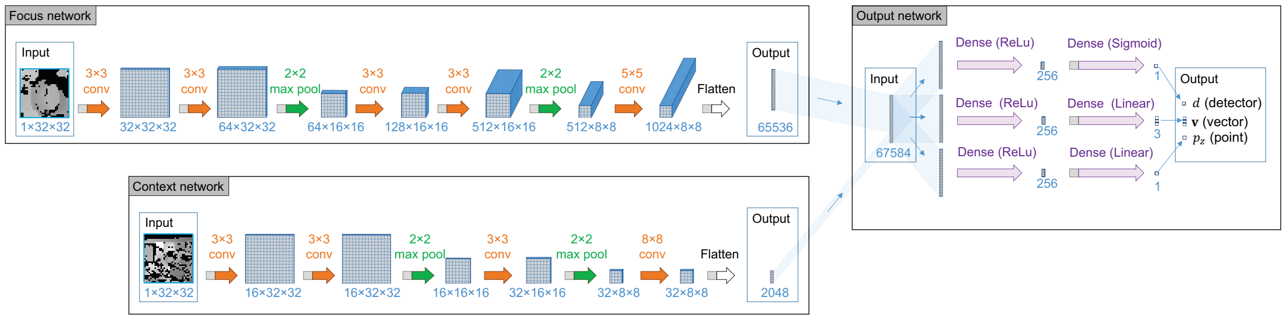

Fig. 3. Architecture of the focus, context and output networks.

III. GRASP POSE DETECTION

We propose the use of a dual-resolution convolutional neural network for estimation of grasps from depth images. Combining a high-resolution with a low-resolution image as input, provides the network with enough information to accurately place grasps on small objects, and enough of an overview of the scene to avoid collisions between the gripper and the local environment.

A. Convolutional Neural Networks

The neural network architecture is illustrated in Fig. 2. An input depth image $\mathbf{I}$ is used to compute a batch of image patches $\mathbf{I}_p$ of size 128×128. The neural network processes each of these image patches independently and produces five outputs for each image patch. The first output d, is the grasp detector confidence and estimates the probability of a valid grasp point in the center of the image patch. At test time, the image patch with the corresponding highest grasp detector confidence is selected, and from this image patch the neural network computes the grasp that the virtual or real robot will perform. Three outputs describe the 3D grasp vector $\mathbf{v}$, which is an approach vector for the grasp point in camera coordinates. The last output p_z, is the point estimator which estimates the distance to the grasp point along the z-axis of the 3D camera, at the x- and y- coordinates of the center of the depth image patch.

From the depth image patch $\mathbf{I}_p$ of size 128×128 pixels, the central 32×32 pixels are considered the focus of the grasp network, and are input to the convolutional focus network $\mathbf{f}_{FN}(\mathbf{x}; \mathbf{W}_{FN})$. A second convolutional network, called the context network $\mathbf{f}_{CN}(\mathbf{x}; \mathbf{W}_{CN})$, takes as input a 4×4 down-sampled version of $\mathbf{I}_p$. The architecture of the focus and context networks are shown in Fig. 3. Each have convolutional layers and max-pooling layers and use the rectified linear unit (ReLu) activation function after each convolutional layer. The vector outputs from these two networks are concatenated and input to the output network $\mathbf{f}_{ON}(\mathbf{x}; \mathbf{W}_{ON})$. The output network has three sub-networks each having two dense layers, with each sub-network computing one of the three outputs d, $\mathbf{v}$, and p_z. The first dense layer in each sub-network uses a (ReLu) activation function. The second dense layer uses a sigmoid activation before the output d, and ReLu before the outputs $\mathbf{v}$ and p_z.

B. Training

The neural network was trained end-to-end, supervised on 400 000 synthetically created training examples. Training was done using the Adam optimizer [18]. One training example consists of a 128×128 image patch input $\mathbf{I}_p$ with its corresponding ground-truth output $\mathbf{y} = \begin{bmatrix} d & \mathbf{v} & p_z \end{bmatrix}$. For false examples, i.e. image patches not containing a grasp, the target vector is set to $\mathbf{y} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \end{bmatrix}$. Given the ground-truth output $\mathbf{y}$, and the output $\hat{\mathbf{y}}$ that the network predicts,

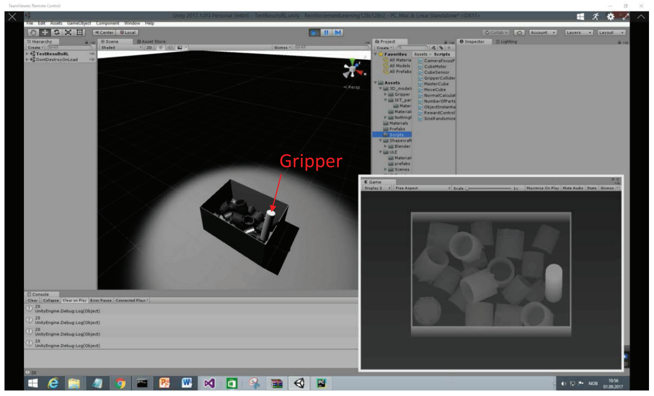

Fig. 4. The simulated environment, used for generating synthetic data set for training the neural network and for evaluating its performance on synthetic depth images.

the cost function is a weighted sum of the detector cost J_d, point regression cost J_{p_z} and the vector regression cost $J_{\mathbf{v}}$, expressed as

$$J_{total}(\mathbf{y}, \hat{\mathbf{y}}) = \alpha J_d(\mathbf{y}, \hat{\mathbf{y}}) + \beta J_{p_z}(\mathbf{y}, \hat{\mathbf{y}}) + \gamma J_{\mathbf{v}}(\mathbf{y}, \hat{\mathbf{y}}). \quad (1)$$

For the grasp detector cost, the cross entropy function is used, giving

$$J_d(\mathbf{y}, \hat{\mathbf{y}}) = -d \ln \hat{d} + (1-d) \ln (1-\hat{d}). \quad (2)$$

For both the point and vector regression costs, the squared error cost function is used. Note that for input training images not containing a valid grasp, we do not have real targets for the point and vector estimators. Therefore, we mask out the point and vector costs for false examples by multiplying with the classification label, giving us the cost functions

$$J_{p_z}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} d (p_z - \hat{p_z})^2 \quad (3)$$

and

$$J_{\mathbf{v}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} d \|\mathbf{v} - \hat{\mathbf{v}}\|_2^2. \quad (4)$$

C. Simulated Environment for Generating Training Data

There is a need for large amounts of data when training deep neural networks and hand labelling of a large data set for the bin picking task would be very time consuming. To avoid this tedious work, we used a simulated environment, shown in Fig. 4. Using the Unity3D game engine and its built-in physics, we created an environment for easy data generation.

The environment is generic and can be used to create data sets for 6 DOF grasping tasks from depth images for any type of rigid object. To generate a data set, a geometric model of the graspable objects needs to be provided. Additional information about weight, reflection and friction coefficients is also needed. Lastly, some valid grasps for each object need to be set. In our experiments we defined 21 preset grasps for each of the three types of graspable metal cylinders. The output from the simulated environment is a depth image and a list of valid grasps in camera coordinates. The data is generated as follows:

1) A random number of parts in the range 1 to 30 are instantiated in mid-air with random orientations.
2) The parts are dropped and allowed to fall to a rest in random positions in the box or on the table.
3) For each preset grasp on each instance of a part, perform a check for collision between the gripper and all other parts and the box. A grasp is a valid grasp if there is no collision.
4) For each instance of a part with at least one valid grasp, only a single valid grasp is selected. The grasp is selected in a manner that favors grasps that are close to the world z-axis and in the direction towards the simulated 3D camera.
5) The simulated 3D camera is randomly rotated and translated before an intensity image $\mathbf{I}_{cam}$ and a depth image $\mathbf{I}_{depth}$ is rendered and saved to disk along with a single valid grasp for each part in the scene.

The neural network requires a data set of 128×128 image patches where each patch either has a grasp in the center, or it has no grasp. The true examples were simply created by cropping patches from the generated depth image, centered around each grasp. An equal amount of false examples were created by cropping random patches from the same image. This way of creating false examples may lead to some false negatives. We assume that the number of possible grasps in the image are outnumbered with a large enough margin by the number of not possible grasps for this to not impact training negatively.

D. Synthetic and Real 3D Camera Images

Depth information is in principle invariant to lighting and texture, which makes it far easier to generate realistic depth images than RGB-images. However, with a real 3D camera, noise and missing depth data occurs even in controlled lighting conditions.

The 3D camera used in the experiments was the Zivid RGB-D camera based on projection of structured light. The camera has high resolution (0.1 mm), high speed (10Hz) and High-dynamic-range (HDR) imaging, which combat the over/under exposure problem, to some degree. However, the captured 3D data still suffers from noise and missing depth data, especially in cluttered scenes with many reflective surfaces, which leads to under exposure in some areas and over exposure in others, as shown to the right in Fig. 5. Additional noise comes as a result of the light from the projector reflecting off multiple objects before reaching the camera.

In order to generate realistic looking depth images, we simulate the missing data in depth images resulting from the reflectivity of the steel parts. We assume that all other objects consist of ideal diffuse reflective surfaces and that the structured light projector is the only light source in the scene. Because the experiments done with the real 3D camera and robot are done in controlled surroundings with controlled lighting conditions, the synthetically created depth images

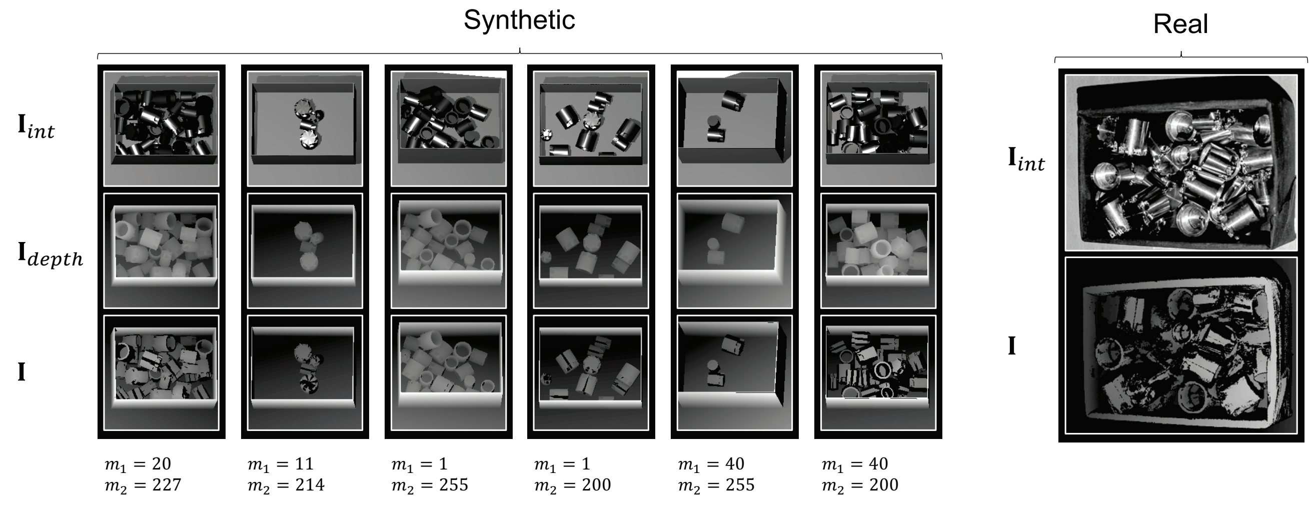

Fig. 5. Synthetic images generated with domain randomization, compared to real image of steel parts in a box. The intensity images $\mathbf{I}_{int}$ are used to mask the depth images $\mathbf{I}_{depth}$ resulting in masked depth images $\mathbf{I}$. The masking is done using randomized masking parameters m_1 and m_2 for each image.

should sufficiently approximate real depth images. Domain randomization over the dynamic range of the depth images is applied to generate training and test data for the neural network. This is done by combining an intensity image patch $\mathbf{I}_{p,int}$ with a depth image patch $\mathbf{I}_{p,depth}$ into an final image patch $\mathbf{I}_p$ defined by

$$\mathbf{I}_p = \mathbf{I}_{p,depth} \cdot \mathbb{I}(\mathbf{I}_{p,int} \geq m_1 \wedge \mathbf{I}_{p,int} \leq m_2) + \epsilon \quad (5)$$

where $\mathbb{I}(\cdot)$ is the indicator function and ϵ is an image of uniformly-distributed random noise satisfying $-1 \leq \epsilon \leq 1$. The dynamic range of the 3D camera is randomly adjusted by setting m_1 and m_2 to uniformly-distributed random numbers satisfying $m_{1,min} \leq m_1 \leq m_{1,max}$ and $m_{2,min} \leq m_2 \leq m_{2,max}$. Additionally, we assume $m_1 \ll m_2$, so that m_1 corresponds to the minimum level of intensity required to provide a valid depth measurement, and m_2 corresponds to the maximum level, i.e. camera saturation. Examples of synthetic images and an example of a real image can be seen in Fig. 5. The synthetic images show the variations due to domain randomization over the following variables of the simulation:

- Number of steel parts
- Size of steel parts
- Position and orientation of steel parts
- Reflectivity of steel parts
- Position and orientation of the 3D camera
- Dynamic range of the 3D camera

In total, this domain randomization is expected to span the range of scenarios sufficiently that a neural net trained on synthetic images will work well on real images.

IV. EXPERIMENTS AND RESULTS

A. Procedure for Grasp Evaluation on Synthetic Data

In order to evaluate the performance of the neural network on a large data set, we tested it on data from a simulation not used during training, using a simplified simulation of a suction gripper. The experiment was conducted as follows:

1) A random number of parts in the range 1-30 were dropped in the box
2) The number of parts present in the scene was noted before the neural network was used to attempt a grasp.
3) If the grasp was successful:
    a) The grasp was logged as successful together with the number of parts present before the grasp was attempted.
    b) The picked part was removed from the scene, allowing the remaining parts to move according to their physics.
    c) If the picked part was the last part in the scene: Go to point 1. Else: Go to point 2.
4) If the grasp was unsuccessful:
    a) The grasp was logged as unsuccessful together with the number of parts present before the grasp was attempted.
    b) The cause of the failure was logged as either: Gripper collision (the virtual gripper collided with the environment or other parts) or poor grasp (outside a set tolerance).
5) Go to point 1.

The experiment in simulation was continued until 12000 parts were picked successfully. The physics of the vacuum gripper was not simulated, instead the criteria for a valid grasp were defined by the position of the gripper on the part and the gripper angle relative to the surface normal on the contact point.

B. Results of Grasp Evaluation on Synthetic Data

The grasp performance of the neural net tested in the simulated environment shows that we achieve an overall success rate of 83 % for the bin picking system, when using

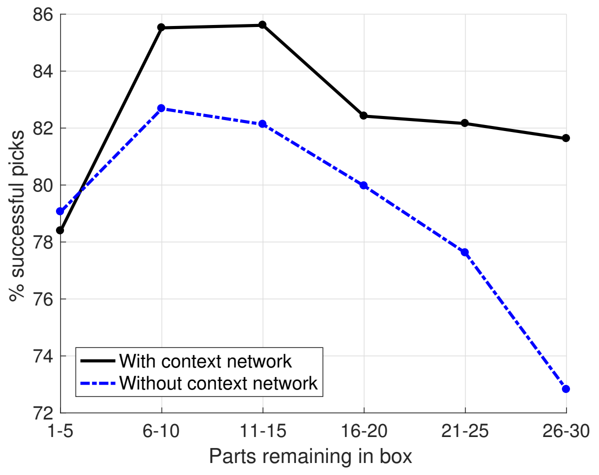

Fig. 6. Successful picks with and without context network, as a function of the number of remaining parts in the box.

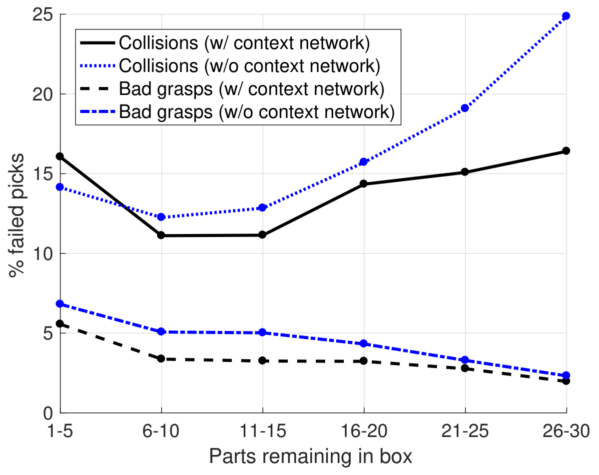

Fig. 7. Failed picks, due to collision and bad grasps, as a function of the number of remaining parts in the box.

the dual-resolution network. The success rate varies as a function of the number of parts remaining in the box (Fig. 6). Most of the errors made were due to collisions between the gripper and the environment. If these errors are ignored, i.e. we evaluate the performance of the system solely on the basis of correctly placed grasps on the steel parts, the success rate is 95 %. In a robot bin picking system this issue would be resolved by implementing a global collision check. It seems to be a trend that the success rate drops when there are few objects left in the box. The likely cause of this is that the network always chooses the most certain grasp in the scene first, leaving the most uncertain grips for last, e.g. difficult grasps like a part in the corner of the box that is only partially viewed by the camera.

C. *Evaluation of the Context Network*

To evaluate the effect of the context network we tested the proposed dual-resolution network, which includes the context network, and a single-resolution network without a context network. The single-resolution network has the same overall architecture as the dual-resolution network, with the difference being that the context network is removed entirely. A separate training was done on the single-resolution network. The single-resolution network without the context network has more picking failures as the number of parts in the box increase (Fig. 7), compared to the dual-resolution network with the context network. The number of collision failures are reduced by up to 34%. When there are few parts in the box there are few local collisions and the context network has little effect. As the number of parts in the box increases, there are potentially more local collision between the gripper and the parts. Overall, the use of a context network in a dual-resolution network can improve grasping success by reducing the number of collisions between the gripper and the parts.

D. *Robot Bin Picking Data and Setup*

For the bin picking experiments with a real robot we used reflective steel parts from an industrial automation application. We used a set of 40 cylindrical parts with diameters varying between 19 mm and 30.7 mm, and lengths varying between 25.7 mm and 31.7 mm. For these experiments, a UR5 6-DOF robotic manipulator from Universal Robots was used. A vacuum gripper with a single suction cup was attached to the end effector. The suction cup had a diameter of 10 mm and could be compressed 4-5 mm in the tool point direction. The design of the gripper puts some physical limitations on the set of grasps that can be executed without collision. Even though the dual-resolution neural network incorporates local collision checking, global collision checking for the whole robot arm was not implemented. This could be solved by running a global collision check (for instance OpenRAVE) on the suggested grasp before execution.

E. *Robot Bin Picking Evaluation Procedure*

A box was filled with steel parts of different sizes in a random manner. An image was captured of the box and used as input to the system, and the grasp with highest score was executed with the robot. If a grasp was unsuccessful, either because of collision or an erroneous grasp estimation, the part was removed manually. Otherwise, the robot removed the object by grasping and picking it up and placing it in a second box. A new image was acquired before attempting to remove a new part. This continued until the box was empty. The robot test was conducted on 6 different boxes, each initialized with 30 parts randomly placed.

F. *Comparing Grasp Placement Performance on Simulated and Real Depth Images*

The grasp placement performance of the dual-resolution network was evaluated on simulated and real depth images. In this evaluation, only the grasp placement is evaluated, and other robot system related errors are not considered. There is good correspondence between the results on simulated and real depth images. The grasp placement performance on real depth images has an overall lower success rate, see Fig. 8. The mean grasp placement success rate for the simulated

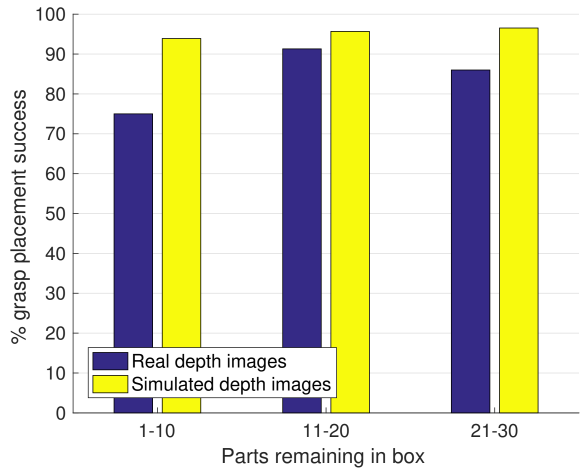

Fig. 8. Grasp placement performance on simulated and real depth images, as a function of the number of remaining parts in the box.

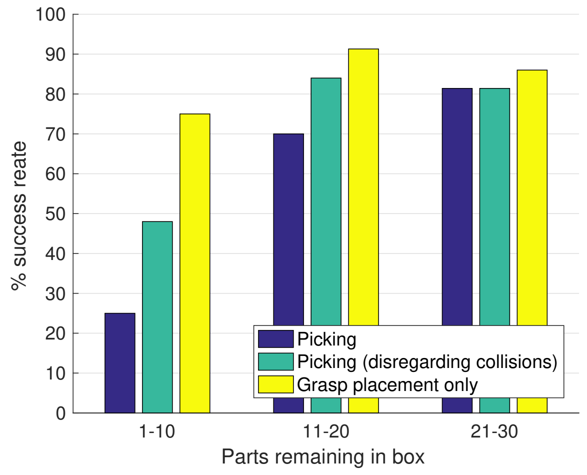

Fig. 9. Real-life bin picking performance, showing the picking success rate with and without disregarding local collisions, as well as grasp placement success.

depth images is 95% and 85% for the real depth images. In both cases there is a decrease in performance towards the bottom of the box, when there are few parts left. However, the performance drop on the real depth images is somewhat higher. This is most likely caused by more missing data in the real depth images, than in the simulated depth images, see Fig. 5.

G. Robot Bin Picking Performance

In the performance evaluation of the robot bin picking system we measure success in several ways, in order to understand the successes and failures of the system. This performance evaluation is summarized in Fig. 9. Picking success is measured by the success rate of the complete robot system performing a grasp. We also measured the picking success that could be achieved by disregarding grasps that could be removed by a global collision check. Finally, we compare this to grasp placement only. Compared to the simulation results, the robot has a very low success rate when picking the last parts in the box. This is mostly due to collisions between the real gripper and environment, and these errors stem from some differences between the physical and the virtual gripper. In addition to collisions, in the robot test, the failed grasps were due to small errors in grasp position or angle, resulting in failed grasps due to lack of suction. This error happened more frequently at the bottom of the box. The last objects are often the ones which are most difficult to grasp (e.g. in a corner), Levine et al. [22] reported a similar trend. Additionally, objects in the bottom of the box have less depth data due to occlusion. Also, parts oriented with the opening facing up occurred more frequently in the real-world test, because of a different weight distribution than in the simulations. Some differences between the real-world and the simulation combined lead to a more challenging test scenario in the real world than in the simulation. For situations when the real-world and the simulated environment are more similar, for instance when there are 21-30 parts in the box, the performance in the real-world and in simulation are equally good (81% and 82% picking success rates respectively, see Fig. 9 and Fig. 6). This result suggests that improvements in the simulation, to more accurately represent the real 3D camera, robot and gripper, will result in higher picking success in the real world.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a method for robotic bin picking of reflective steel parts. We proposed a dual-resolution convolutional neural network for 5-DOF grasping, trained entirely by domain randomization in a simulated environment. The system was tested on simulated data as well as in the real-world using a robot with a 5-DOF placement vacuum gripper. Using a context network improves the grasp performance by minimizing collisions between the gripper and the local environment. We demonstrate that training of the neural net by domain randomization on a large set of synthetic depth images is an effective and useful approach when the simulated environment closely resembles the real world. Future work will focus on more accurately simulating the 3D camera to improve the model of missing data in the depth images, and implementing a more accurate model of the gripper. We will also explore how we can improve the simulated data through Adversarial Networks.

ACKNOWLEDGMENT

This project was funded by the Research Council of Norway through the SFI Manufacturing Centre for Research-based Innovation, and in part from Research Council of Norway Grant No. 262900.

REFERENCES

[1] Gualtieri, M., ten Pas, A., Saenko, K., and Platt, R. (2016,October). High precision grasp pose detection in dense clutter. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on (pp. 598-605). IEEE.

[2] Kappler, D., Bohg, J., and Schaal, S. (2015, May). Leveraging big data for grasp planning. In Robotics and Automation (ICRA), 2015 IEEE International Conference on (pp. 4304-4311). IEEE.

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

[4] Redmon, J., and Angelova, A. (2015, May). Real-time grasp detection using convolutional neural networks. In Robotics and Automation (ICRA), 2015 IEEE International Conference on (pp. 1316-1322). IEEE.

[5] Pinto, L., and Gupta, A. (2016, May). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (pp. 3406-3413). IEEE.

[6] Saxena, A., Driemeyer, J., and Ng, A. Y. (2008). Robotic grasping of novel objects using vision. The International Journal of Robotics Research, 27(2), 157-173.

[7] Jiang, Y., Moseson, S., and Saxena, A. (2011, May). Efficient grasping from rgbd images: Learning using a new rectangle representation. In Robotics and Automation (ICRA), 2011 IEEE International Conference on (pp. 3304-3311). IEEE.

[8] Bicchi, A., and Kumar, V. (2000). Robotic grasping and contact: A review. In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on (Vol. 1, pp. 348-353). IEEE.

[9] Miller, A. T., Knoop, S., Christensen, H. I., and Allen, P. K. (2003, September). Automatic grasp planning using shape primitives. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on (Vol. 2, pp. 1824-1829). IEEE.

[10] Miller, A. T., and Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. IEEE Robotics and Automation Magazine, 11(4), 110-122.

[11] Pelossof, R., Miller, A., Allen, P., and Jebara, T. (2004, April). An SVM learning approach to robotic grasping. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on (Vol. 4, pp. 3512-3518). IEEE.

[12] Len, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., ... and Dillmann, R. (2010, November). Opengrasp: a toolkit for robot grasping simulation. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots (pp. 109-120). Springer Berlin Heidelberg.

[13] Huang, P. C., Lehman, J., Mok, A. K., Miikkulainen, R., and Sentis, L. (2014, December). Grasping novel objects with a dexterous robotic hand through neuroevolution. In Computational Intelligence in Control and Automation (CICA), 2014 IEEE Symposium on (pp. 1-8). IEEE.

[14] Pas, A. T., and Platt, R. (2015). Using Geometry to Detect Grasps in 3D Point Clouds. arXiv preprint arXiv:1501.03100.

[15] Molchanov, P., Gupta, S., Kim, K., and Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-7).

[16] Zeng, A., Yu, K. T., Song, S., Suo, D., Walker Jr, E., Rodriguez, A., and Xiao, J. (2016). Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. arXiv preprint arXiv:1609.09475.

[17] Goodale, M. A., and Milner, A. D. (1991). A neurological dissociation between perceiving objects and grasping them. Nature, 349(6305), 154.

[18] Kingma, D., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[19] Domae, Y., Okuda, H., Taguchi, Y., Sumi, K., and Hirai, T. (2014, May). Fast graspability evaluation on single depth maps for bin picking with general grippers. In Robotics and Automation (ICRA), 2014 IEEE International Conference on (pp. 1997-2004). IEEE.

[20] Mahler, J., Pokorny, F. T., Hou, B., Roderick, M., Laskey, M., Aubry, M., ... and Goldberg, K. (2016, May). Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (pp. 1957-1964). IEEE.

[21] Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., and Goldberg, K. (2017). Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. arXiv preprint arXiv:1703.09312.

[22] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research, 0278364917710318.

[23] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. arXiv preprint arXiv:1703.06907.

[24] Abbeloos, W., and Goedem, T. (2016, June). Point Pair Feature based Object Detection for Random Bin Picking. In Computer and Robot Vision (CRV), 2016 13th Conference on (pp. 432-439). IEEE.

[25] Ellekilde, L. P., Jorgensen, J. A., Kraft, D., Kruger, N., Piater, J., and Petersen, H. G. (2012, October). Applying a learning framework for improving success rates in industrial bin picking. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 1637-1643). IEEE.

[26] Buchholz, D., Kubus, D., Weidauer, I., Scholz, A., and Wahl, F. M. (2014, May). Combining visual and inertial features for efficient grasping and bin-picking. In Robotics and Automation (ICRA), 2014 IEEE International Conference on (pp. 875-882). IEEE.

[27] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010, June). Model globally, match locally: Efficient and robust 3D object recognition. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 998-1005).

[28] Schwarz, M., and Behnke, S. Data-efficient Deep Learning for RGB-D Object Perception in Cluttered Bin Picking. In Warehouse Picking Automation Workshop (WPAW), IEEE International Conference on Robotics and Automation (ICRA), 2017.

[29] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017, July). Learning from simulated and unsupervised images through adversarial training. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 3, No. 4, p. 6).