PEGG-Net: Background Agnostic Pixel-Wise Efficient Grasp Generation Under Closed-Loop Conditions

Haozhe Wang¹²*, Zhiyang Liu¹*, Lei Zhou¹, Huan Yin³, and Marcelo H. Ang Jr.¹

Abstract—Performing closed-loop grasping at close proximity to an object requires a large field of view. However, such images will inevitably bring large amounts of unnecessary background information, especially when the camera is far away from the target object at the initial stage, resulting in performance degradation of the grasping network. To address this problem, we design a novel PEGG-Net, a real-time, pixel-wise, robotic grasp generation network. The proposed lightweight network is inherently able to learn to remove background noise that can reduce grasping accuracy. Our proposed PEGG-Net achieves improved state-of-the-art performance on both Cornell dataset (98.9%) and Jacquard dataset (93.8%). In the real-world tests, PEGG-Net can support closed-loop grasping at up to 50Hz using an image size of 480×480 in dynamic environments. The trained model also generalizes to previously unseen objects with complex geometrical shapes, household objects and workshop tools and achieved an overall grasp success rate of 91.2% in our real-world grasping experiments.

Index Terms—RGB-D Perception, Perception for Grasping and Manipulation, Background Agnostic Grasping, Pixel-Wise Grasp Synthesis, Closed-Loop Grasping

I. INTRODUCTION

Grasping is an essential operation for robotic manipulation. Vision-based robotic grasping aims to use input images from cameras or depth sensors to grab and lift objects using grippers. With the development of recent deep learning techniques, many research works [1][2][3] have proposed methods to achieve object independent robotic grasping.

Closed-loop grasping using the eye-in-hand approach requires a large field-of-view (FOV) to be able to see the target object in totality even when the gripper is at close proximity. The grasp pose predicted by the network may not be precise in the final stage of the grasping process if there is only a partial view of the object. However having a large field-of-view would mean that background interference could cause grasp predictions to be inaccurate and unstable initially when the gripper is still far away from the object.

The way to solve this dilemma is by designing a network that can learn to ignore distractions in the background. Many previous works have focused on designing networks with a limited field of view [2][3][4][5][6][7]. Images from the Cornell grasping dataset [8] are first center-cropped to a small size and are then used to train the grasp generation network. Doing so will first remove any background distractions in the images and then allow the network to focus on learning the distinct object features and achieve very high accuracy on the Cornell dataset. However, training a grasp prediction network that does not take account the background environment is not practical. In the real world, parts of the background environment that appear in the input frame will negatively influence the decisions made by the network and drastically reduce its grasping confidence and also the gripper orientation and width predictions. Our empirical observations from reproducing previous works by Morrison et al. [2][6] and Kumra et al. [3] show that their networks are unable to ignore the influence of the background and will sometimes predict invalid grasping points that are in the background. We provide a qualitative comparison between our work and those by Morrison et al. and Kumra et al. in Fig. 2.

Based on the analyses above, we conclude there are research gaps in vision-based robotic grasping. This paper aims to address these research gaps. In this paper, our goal is to design a lightweight network that can predict the 2D planar grasp of a novel object without prior knowledge of the pose of the object itself. At the same time, it must have the ability to remove redundant background features. The two main contributions of this paper can be summarized as follows:

1) We propose PEGG-Net, a novel pixel-wise grasp generation convolutional neural network (CNN) architecture that can learn to ignore background features in a scene. This results in significant improvements in grasping accuracy using either depth or RGB-D
When evaluated on publicly available grasping datasets, PEGG-Net achieved better than state-of-the-art accuracy of 98.9% and 93.8% on the Cornell and Jacquard grasping datasets respectively. We also demonstrate the robustness and generalization capabilities of PEGG-Net and our closed-loop control system with real-world grasping experiments on Kinova Movo. We tested our grasping system using objects with adversarial geometry from the DexNet 2.0 dataset [9], household objects, and workshop tools. All objects used in our real-world test were unseen during training stage. We ran our experiments under static, dynamic, and cluttered conditions and achieved higher levels of accuracy compared to previous works.

Overall, PEGG-Net together with our fully closed-loop control system achieves significant improvements in grasp stability, accuracy and generalization capability compared to previous works in this area. Our work enhances the distinct advantages of a closed-loop planar grasping system without introducing new limitations.

II. RELATED WORKS

A. Deep Learning Methods for Grasping Novel Objects

Deep learning has been instrumental in advancing the field of computer vision in general and hence vision-based robotic grasping as well. Many recent works applying deep learning techniques to grasping novel objects have achieved great success.

Regression-based methods proposed by Redmon and Angelova [7] and Kumra et al. [10] use a CNN to regress a single best grasp pose from an input image with only one object in it. However, such networks are unable to handle cluttered environments and may generate an invalid grasp that is the average of the possible grasps for an object. Chu et al. [4] proposed a CNN that can predict multiple grasps for multiple objects simultaneously. However, the network is too computationally intensive. Previous works have also proposed many methods of grasp prediction using multimodal data [3][10][11][12]. Other deep learning techniques [1][8][9][11][13][14][15] first classify grasp candidates sampled from an image or point cloud, then rank them individually using CNNs to determine the best grasp pose.

It is important to note that most grasp prediction networks proposed in recent works have a large number of parameters (in the order of tens of millions to even hundreds of millions of parameters) [4][8][9][14][16], which makes them very computationally intensive. Therefore most networks proposed in previous works can only be executed in an open-loop manner. The critical disadvantage of open-loop grasping systems is that they are unable to grasp objects in dynamic environments, for example, objects moving on a conveyor belt. Even if an object is static, sensor and actuation errors can also affect the grasping accuracy of open-loop grasping systems [8]. Hence, we argue that adding closed-loop grasping system is a more robust solution for many practical applications.

B. Visual Servoing for Closed-Loop Grasping

Visual servoing enables robots to perform grasping in dynamic environments and without the need for fully accurate camera calibration or position control. Unlike open-loop grasping, closed-loop grasp allows for grasp poses that can be continuously refined as the gripper approaches an object. In order to achieve fast and accurate closed-loop grasping, the grasp prediction network must be able to execute in real time (at least as fast as the sampling rate of the camera), and there should be minimal delay in the control system so that the robot can respond immediately to the changing environment.

Few previous works [2][8][17][18] have integrated visual servoing techniques into their robotic grasping systems. CNN-based controllers proposed in recent years [17][18] combine deep learning with closed-loop grasping. However, the controllers used in those works can run no faster than 5Hz, which is too slow to support closed-loop grasping in dynamic environments. However, PBVS has achieved closed-loop grasping at up to 50Hz [2], much faster compared to CNN-based methods. Therefore, we combine PBVS with PEGG-Net in our closed-loop robotic grasping system.

III. PROBLEM FORMULATION

For PEGG-Net to predict a 2D planar grasp on an arbitrary object, we adopt a grasp definition similar to other related works [2][3]. Using a RGB-D camera with known intrinsic parameters and given the transformation relationship between the camera and the end-effector, PEGG-Net is able to generate a grasp in the image coordinate frame, which is defined as:

\[ \hat{g} = (\hat{p}, \hat{\phi}, \hat{\omega}, \hat{q}) \]  

(1)

where \( \hat{p} = (u, v) \) are the 2D coordinates of the grasping point in the image frame, \( \hat{\phi} \) is the rotation angle around the vertical orientation of the image, \( \hat{\omega} \) is the grasping width in pixels and \( \hat{q} \) is the grasp quality (the probability of a successful grasp). It is worth mentioning here that our grasping width is not fixed, which is unlike other works [16][18].

To execute the grasp predicted by PEGG-Net, the grasp \( \hat{g} \) in the image coordinate frame must be converted to the corresponding grasp \( g \) in the world coordinate frame, which is defined as:

\[ g = (p, \phi, \omega, q) \]  

(2)

where \( p = (x, y, z) \) in world coordinates \((x, y)\) is the center of the gripper and \( z \) is the depth of the grasp), \( \phi \) is the gripper’s rotation angle about the z-axis, \( \omega \) is the gripper’s opening width \( \omega \), and \( q \) is the grasp quality. \( \hat{g} \) and \( g \) are governed by the following relationship:

\[ g = T_{RC}(T_{CI}(\hat{g})) \]  

(3)

Where \( T_{CI} \) is the transform from the image coordinate frame to the camera coordinate frame and \( T_{RC} \) is the transformation from the camera coordinate frame to the world coordinate frame. Using (3), we can compute the grasp pose in the real-world.
In our work, PEGG-Net predicts a grasp for every pixel in an image. Therefore, we can apply (1), (2), and (3) to an image with multiple grasps. The set of all possible grasps in the image coordinate frame can be denoted as:

\[ \hat{G} = (\hat{\Phi}, \hat{\Omega}, Q) \in \mathbb{R}^{3 \times H \times W} \]  

(4)

where

\[ \mathbb{R}_{\geq 0} = \{ x \in \mathbb{R} : x \geq 0 \} \]  

(5)

\[ \hat{\Phi} = \{ \hat{\phi} \in \mathbb{R}_{\geq 0} \}, \hat{\phi} \in \mathbb{R}^{H \times W} \]  

(6)

\[ \hat{\Omega} = \{ \hat{\omega} \in \mathbb{R}_{\geq 0} \}, \hat{\omega} \in \mathbb{R}^{H \times W} \]  

(7)

\[ Q = \{ q \in \mathbb{R}_{\geq 0} \}, Q \in \mathbb{R}^{H \times W} \]  

(8)

and \( \hat{\Phi}, \hat{\Omega} \) and \( Q \) are the feature embeddings \( (H \times W) \) for \( \hat{\phi}, \hat{\omega} \) and \( q \) respectively.

This is a more computationally efficient way of formulating the grasping problem, as there is no need to sample and rank grasp candidates [1][8][9][11][14][15]. We can simply obtain the best grasp in the image coordinate frame by computing \( g_{\text{best}} = \max \hat{G} \) and obtain the best grasp in the world coordinate frame \( g_{\text{best}} \) using (3).

IV. INTERGRATED CLOSED-LOOP ROBOTIC GRASPING SYSTEM

A. Network Architecture

Our network is a fully convolutional encoder-decoder network, shown in Fig. 3. The encoder uses residual blocks and strided convolutions to extract key features of graspable objects in the workspace and learns to ignore the background. The output from the encoder network is passed through a modified spatial pyramid pooling (SPP) module proposed by Huang and Wang in [19]. The purpose of the SPP module is to extract the multi-scale local region features extracted by the encoder using max-pooling kernels of different sizes. We used max-pooling kernels of size \( 5 \times 5 \), \( 9 \times 9 \), and \( 13 \times 13 \). The decoder, which was implemented using the sub-pixel convolution method [20], reconstructs the object features extracted by the encoder. Skip connections are added between the encoder and decoder layers to allow the decoder to use features extracted at various stages in the encoder to reconstruct the object features. In all convolution blocks except the last one, which uses the ReLU activation function, we use the Mish [21] activation function and add batch normalization to all layers. The Mish function is a smooth curve, and smooth activation functions allow for better information propagation deeper into the neural network, and thus better accuracy and generalization. The network outputs a grasp confidence heatmap, a grasp angle map, and a gripper width map. The location of the pixel with the highest grasp confidence is selected from the grasp confidence heatmap, and the corresponding gripper width and angle at that same location is used to set the grasp pose of the gripper.

We conduct an ablation study to determine the combination of downsampling method, activation function, and loss function to determine the combination which gives the highest accuracy. Our results are summarized in Table I. Our final network has 1.38 million parameters. With the exception
Grasp Quality Heatmap

| Combination                                | Accuracy D (%) | Accuracy RGB (%) |
|--------------------------------------------|----------------|------------------|
| Max-Pool + Mish + Smooth L1 loss           | 51.7           | 93.3             |
| Strided Conv. + ReLU + Smooth L1 loss      | 71.9           | 91.0             |
| Strided Conv. + Mish + MSE loss            | 62.9           | 92.1             |
| Strided Conv. + Mish + Smooth L1 loss      | **87.6**       | **93.3**         |

TABLE II: Network size comparison of different grasping methods

| Author       | Method         | Number of Parameters (Approx.) |
|--------------|----------------|-------------------------------|
| Morrison [6] | GGCNN2         | **66 k**                      |
| Kumra [3]    | GR-ConvNet     | 1.9 million                   |
| Cao [12]     | Efficient Grasping | 4.67 million              |
| Mahler [9]   | DexNet 2.0     | 16 million                    |
| Pinto [14]   | AlexNet        | 61 million                    |
| Chu [4]      | ResNet-50 + GPN| 216 million                  |
| Ours         | PEGG-Net       | **1.38 million**             |

Fig. 3: The Architecture of PEGG-Net.

TABLE I: Ablation Study on the combination of downsampling method, activation function, and loss function

B. Closed-Loop Control System

We implemented a PBVS controller [22] to perform closed-loop grasping. We move the end-effector to the final grasping position simply by controlling its velocity. Our integrated robotic grasping system can easily achieve closed-loop control at up to 50Hz, enabling fast and accurate grasping in highly dynamic environments. The RealSense camera is initially 50cm above the workspace. Depth images are generated at a rate of 30Hz and processed by PEGG-Net in real time. There may be many grasping points on the object with similar grasp quality scores. To avoid switching rapidly between grasping points and confusing the controller, we compute 5 grasps from the highest local maxima of \( \hat{G} \) and select which is closest (in the image coordinate frame) to the grasp location predicted by PEGG-Net in the previous iteration. Similar to the method used in [2], the controller publishes a 6D velocity signal \( \vec{v}_{EE} \) to the ROS topic controlling the motion of the end-effector at a frequency of 30Hz, much faster compared to the movement of the robotic arm. Therefore, there is unlikely to be a major change between frames. The system is initialized to track the global maxima of \( \hat{Q} \) at the beginning of each grasp attempt. PEGG-Net and the PBVS controller from a feedback loop, which allows us to perform grasping in dynamic environments. The gripper fingers are controlled via velocity signals sent from the task controller. Control is stopped when the grasp pose is reached. A grasp attempt is considered to be successful if the object can be successfully lifted to the dropping position outside the workspace.

V. EXPERIMENTS

A. Training Strategy

PEGG-Net was trained on a NVIDIA Tesla V100 GPU with 16GB of memory. We trained 23 PEGG-Net models using images of various sizes and modalities, of which 20 were trained on the Cornell grasping dataset and 3 were trained on the Jacquard grasping dataset. For both Cornell and Jacquard datasets, we used 90% of the images for training and 10% for evaluation. We also augmented both datasets by using random crops and zooms. The network was trained using the Adam optimizer [23] with a batch size of 8.
TABLE III: Summary of Antipodal Grasping Datasets

| Dataset     | Modality | N_{objs} | N_{imgs} | N_{grasps} |
|-------------|----------|----------|----------|------------|
| Cornell [8] | RGB-D    | 240      | 885      | 8019       |
| Jacquard [24] | RGB-D    | 11k      | 54k      | 1.1M       |
| DexNet 2.0  | D        | 1500     | 6.7M     | 6.7M       |

B. Evaluation on Public Datasets

Table III shows a summary of the publicly available antipodal grasping datasets. $N_{objs}$, $N_{imgs}$, and $N_{grasps}$ are the number of objects, images, grasps respectively in each dataset. We used the Cornell and Jacquard datasets for training and evaluating our model. We used the adversarial objects from the DexNet 2.0 dataset in our real-world grasping experiments.

To compare our results with previous works trained on the Cornell and Jacquard grasping datasets, we use the rectangle metric proposed by Jiang et al. in [13]. The rectangle metric states that a valid grasp prediction must satisfy the following two conditions:

1) The intersection over union (IoU) score between the ground truth grasp rectangle and the predicted grasp rectangle is more than 25%.

2) The offset between the rotation angle $\phi$ of the ground truth grasp rectangle and the predicted grasp rectangle is less than 30°.

Table IV shows the performance of PEGG-Net on the Cornell dataset for different modalities and input image sizes. We compare the accuracy of PEGG-Net with that of 12 previous works. With the exception of [2], [3], [4], and [6], all other previous works do not include open-sourced code. Therefore, we used the results stated in the literature of all 12 previous works for comparison. The ‘-’ in Table IV indicates that the data could not be found in the literature. All input images used for evaluating our network have been center-cropped from the original image size of 640 $\times$ 480 in the Cornell dataset to the sizes listed in Table IV. When evaluated on an object-wise split, our network outperforms almost all other networks for each modality and image-size. This is an important breakthrough as we expect PEGG-Net to be deployed to grasp objects unseen during training. Therefore, good performance on an object-wise split is more meaningful compared to that on an image-wise split. PEGG-Net achieves remarkable performance even when we use images of size 480 $\times$ 480, which includes the background distractions that are present in each image in the Cornell dataset. This demonstrates that PEGG-Net has learnt to ignore the background in the training process. It is important to note that PEGG-Net has achieved this result despite only being a small fraction of the size of many recently proposed networks. This is the key reason why our grasping network can support closed-loop grasping in dynamic environments.

Table V shows the performance of PEGG-Net for different modalities compared to previous networks proposed by Morrison et al. [2][6] and Kumra et al. [3]. Although PEGG-Net is slightly less accurate than GR-ConvNet, this does not affect its real-world grasping performance. From Table VI, it can be observed that PEGG-Net can still achieve better real-world grasping performance than the vast majority of other competitive networks (most of them much larger than PEGG-Net), including GR-ConvNet. For closed-loop robotic grasping, achieving the best trade-off between speed and accuracy is crucial. PEGG-Net has approximately 27% fewer parameters but no more than 4% less accurate than GR-ConvNet. Therefore PEGG-Net achieves a better trade-off between speed and accuracy compared to GR-ConvNet.

C. Set-up in the Real-world

For our real-world experiments, we use PEGG-Net trained on the Jacquard dataset [24] using input images of size 480 $\times$ 480. Based on the real-time inference results of PEGG-Net, closed-loop robotic grasping is conducted on a 7-DoF robotic arm of the Kinova Movo mobile manipulator fitted with a Kinova KG-Series 3-fingered gripper. An Intel RealSense L515 LiDAR camera is mounted on its wrist, as shown in Fig. 5(a), positioned approximately 8cm above the closed fingertips. We ensure that the fingertips are perpendicular to the table. The hand-eye calibration method is used to calibrate this setup before each experiment. From Fig. 5(b), it can be observed that the poster and white box upon which the mustard bottle rests is ignored by PEGG-Net and only the mustard bottle can be seen in the heatmap shown in Fig. 5(c). This shows that PEGG-Net indeed has the ability to remove background influence and only focus on the graspable objects in the workspace.

All computations of the grasp generation network were performed on a computer running Ubuntu 16.04 with a 3.7 GHz AMD Ryzen 5900X CPU and NVIDIA GeForce RTX 3080 GPU. On this platform, our network takes 10ms to compute a single RGB-D frame, and the entire grasping pipeline (including all pre-processing and post-processing computations) takes 20ms (50Hz).

1) Hardware Limitations:

1) We center-crop the RGB and depth image to 480 $\times$ 480 for grasp prediction. Using this image resolution, we are able to maintain closed-loop control up to the point where the height of the camera is 18cm from the object. Below this height, the RealSense camera is unable to provide any depth measurements. At this height, the tip of the end-effector is only 10cm away from the object.

2) The wavelength of the laser used by RealSense L515 LiDAR is close to that of infrared light from the sunlight, which can degrade the quality of depth images owing to the interference under outdoor or uncontrolled lighting conditions.

3) In theory, a three-fingered gripper can provide extra dexterity and increase the amount of friction between the gripper and the object, making the grasps more stable. However, we think that it would be unfair to use a three-fingered gripper to compare the grasping success rate of our model with that of previous works.

| Dataset     | Modality | N_{objs} | N_{imgs} | N_{grasps} |
|-------------|----------|----------|----------|------------|
| Cornell     | RGB-D    | 240      | 885      | 8019       |
| Jacquard    | RGB-D    | 11k      | 54k      | 1.1M       |
| DexNet 2.0  | D        | 1500     | 6.7M     | 6.7M       |
Therefore, we removed one finger and only use the remaining two fingers that are parallel to each other.

D. Test Objects for Closed-loop Grasping in the Real-world

We evaluate the performance of our network using three sets of objects: objects with adversarial geometry, household objects, and workshop tools. Fig. 4 shows all the objects that were used during the experiments. We conduct a set of three tests for each set of objects: a static grasping test, a dynamic grasping test, and a cluttered grasping test. During the static grasping test, each object was tested in isolation for 10 different positions, orientations, and poses. For the dynamic grasping test, each object was tested in isolation and moved randomly during each of 10 grasp attempts. Following the technique used by Morrison et al. in [2] to assist reproducibility, we translate each object by at least 10 cm and rotate it by at least 25°. For the cluttered grasping test, we place all the object of a set into the workspace and count the number of grasp attempts needed to clear all objects from the workspace. We divide the number of objects in the set by the number of grasp attempts needed to clear all objects from the workspace to obtain the grasp success rate.

1) Objects with adversarial geometry: This set consists of 13 3D-printed objects with adversarial geometry, which were used by Mahler et al. [9] to evaluate the performance of GQ-CNN. All objects in this set have complex geometrical features. Even slight errors in the grasp prediction will result in a grasp failure. Therefore, this set of objects is very suitable for verifying the accuracy of the grasp poses generated by PEGG-Net. To the best of our knowledge, besides Mahler et al. [9], we are the only one to evaluate our network on all 13 adversarial objects in the DexNet 2.0 dataset for 2D planar grasping.

2) Household Objects: This set consists of 14 common household items of varying sizes and shapes. To minimize redundancy and fairly evaluate the generalizability of our network to novel objects, none of the objects in this set have similar shapes.

3) Workshop Tools: The objects in this set have been specially selected to challenge our network. From the perspective of a depth image, all of the workshop tools in this set have depths that are very close to the mean depth of the image. Using the mean-thresholding method to remove the background in the input depth image would result in the object being barely visible to the network. This is a quintessential problem that can be solved by PEGG-Net. Any background features present in the workspace would have to be thoroughly filtered out by PEGG-Net in order to accurately predict the grasp pose of the tools. Accurate depth perception is also important to prevent the gripper from colliding with the object or the surface of the workspace.

E. Real-world experiment results

Table VI compares our real-world experiment results to that of previous works under static, dynamic and cluttered conditions. Note that household objects are an extremely large category of objects, and previous works have each used different household objects for evaluating their networks. Therefore, the comparative performance for household objects listed in Table VI is only indicative. PEGG-Net performs on par with GQ-CNN (which was trained on the adversarial objects) despite ours not being trained on the adversarial objects. PEGG-Net has also performed well when grasping the workshop tools, which indicates that it is indeed able to ignore redundant background information and generate accurate grasp predictions. We compute the overall grasp success rate by dividing the total number of successful grasp attempts by the total number of grasp attempts over all the grasp experiments we conducted. Overall, we have achieved a remarkable grasp success rate of 91.2%, and PEGG-Net has achieved both a qualitative and quantitative breakthrough compared to previous works.

VI. CONCLUSIONS

In this paper, we presented PEGG-Net, a novel real-time, pixel-wise grasp detection model that can learn to ignore background interference. We evaluated our network on the Cornell and Jacquard grasping datasets and also performed real-world grasping experiments with a robotic arm and using objects unseen during training. Both the benchmark results on publicly available grasping datasets and the results from the real-world grasping experiments demonstrate the
TABLE IV: Detection accuracy on Cornell Dataset

| Author     | Method                               | Modality | Input Size | Accuracy       | Speed (ms) | Closed-loop |
|------------|--------------------------------------|----------|------------|----------------|------------|-------------|
|            |                                      |          |            | Image-wise (%) | (%)        |             |
|            |                                      |          |            | Object-wise (%)| (%)        |             |
| Lenz [8]   | SAE, struct. reg. two-stage (running on CPU) | RGB-D    | 300 × 200  | 73.9           | 75.6       | 13500       |
| Redmon [7] | AlexNet, Multi-grasp                 | RGB-D    | 224 × 224  | 88.0           | 87.1       | 76          |
|            |                                      |          |            |                |            |             |
|            |                                      |          |            |                |            |             |
|            |                                      |          |            |                |            |             |
| Kumra [10] | ResNet-50                            | RGB-D    | 224 × 224  | 89.2           | 88.9       | 103         |
| Zhang [5]  | Multi-modal Fusion                   | RGB-D    | 224 × 224  | 88.9           | 88.2       | 117         |
| Asif [25]  | GraspNet                             | RGB-D    | 224 × 224  | 90.6           | 90.2       | 24          |
| Asif [26]  | EnsembleNet                          | RGB-D    | 224 × 224  | -              | 93.7       | -           |
| Kumra [3]  | GR-ConvNet                           | D        | 224 × 224  | 93.2           | 94.3       | 19          |
|            |                                      | RGB-D    | 224 × 224  | 97.7           | 96.6       | 20          |
| Chu [4]    | ResNet-50 + GPN + RoI Pooling        | RGB-D    | 227 × 227  | 96.0           | 96.1       | 120         |
| Morrison [2]| GGCNN                               | D        | 300 × 300  | -              | 76.4       | 19          |
| Morrison [6]| GGCNN2                              | D        | 300 × 300  | -              | 66.3       | 20          |
| Cao [12]   | Efficient Grasping                   | D        | 300 × 300  | 98.9           | 95.5       | 6           |
|            |                                      | RGBD     | 300 × 300  | 98.9           | 97.8       | 6           |
| Wang [27]  | GPWRG                                | D        | 400 × 400  | 94.4           | 91.0       | 8           |
| Ours       | PEGG-Net                             | D        | 224 × 224  | 92.1           | 94.4       | 3.1         |
|            |                                      |          |            | 93.3           | 98.9       | 3.1         |
|            |                                      | D        | 304 × 304  | 88.8           | 93.3       | 3.8         |
|            |                                      |          |            | 93.3           | 93.3       | 3.8         |
|            |                                      | D        | 320 × 320  | 87.6           | 94.4       | 4.3         |
|            |                                      |          |            | 92.1           | 96.6       | 4.3         |
|            |                                      | D        | 400 × 400  | 92.1           | 86.5       | 6.7         |
|            |                                      |          |            | 94.4           | 87.6       | 6.7         |
|            |                                      | D        | 480 × 480  | 92.1           | 87.6       | 10          |
|            |                                      |          |            | 95.5           | 91.0       | 10          |

detection accuracy and grasping robustness of the proposed approach. Our network is fast enough to support closed-loop grasping at up to 50Hz in dynamic environments.

In our future work, we would like to extend our solution to object independent 6-DoF robotic grasping using different types of grippers and also for grasping deformable objects and reflective objects, which are also commonly seen in daily household items.

VII. ACKNOWLEDGMENT

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

REFERENCES

[1] A. Zeng, S. Song, K.-T. Yu, E. Donlson, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Daffe, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching,” The International Journal of Robotics Research, vol. 0, no. 0, p. 0278364919868017, 0. [Online]. Available: https://doi.org/10.1177/0278364919868017
[2] D. Morrison, P. Corke, and J. Leitner, “Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach,” in Proc. of Robotics: Science and Systems (RSS), 2018.
[3] S. Kumra, S. Joshi, and F. Sahin, “Antipodal robotic grasping using generative residual convolutional neural network,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 9626–9633.
[4] F.-J. Chu, R. Xu, and P. A. Vela, “Real-world multiobject, multigrasp detection,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3355–3362, 2018.
| Scene  | Methods       | Household objects (%) | Adversarial Objects (%) | Workshop tools (%) |
|--------|---------------|------------------------|-------------------------|--------------------|
| Static | Morrison [2]  | 92.0                   | 84.0                    | -                  |
|        | Kumra [3]     | 95.4                   | -                       | -                  |
|        | Lenz [8]      | 89                     | -                       | -                  |
|        | Mahler [9]    | 80.0                   | 93.0                    | -                  |
|        | Pinto [14]    | 73.0                   | -                       | -                  |
|        | Chu [4]       | 89.0                   | -                       | -                  |
|        | Viereck [18]  | 98.0                   | -                       | -                  |
|        | Ours          | 96.4                   | 93.1                    | 87.0               |
|        |               | (135/140)              | (121/130)               | (87/100)           |
| Dynamic| Morrison [2]  | 88.0                   | 83.0                    | -                  |
|        | Ours          | 92.8                   | 90.0                    | 85.0               |
|        |               | (130/140)              | (117/130)               | (85/100)           |
| Clutter| Morrison [2]  | 87.0                   | -                       | -                  |
|        | Viereck [18]  | 89.0                   | -                       | -                  |
|        | Ours          | 88.3                   | 92.3                    | 100.0              |
|        |               | (15/17)                | (12/13)                 | (10/10)            |

[5] Q. Zhang, D. Qu, F. Xu, and F. Zou, “Robust robot grasp detection in multimodal fusion,” MATEC Web of Conferences, vol. 139, p. 00060, 01 2017.

[6] D. Morrison, P. Corke, and J. Leitner, “Learning robust, real-time, reactive robotic grasping,” The International Journal of Robotics Research, vol. 39, pp. 183 – 201, 2020.

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

[8] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” International Journal of Robotics Research, vol. 34, 01 2013.

[9] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” arXiv preprint arXiv:1703.09312, 2017.

[10] S. Kumra and C. Kanan, “Robotic grasp detection using deep convolutional neural networks,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 769–776, 2017.

[11] Z. Wang, Z. Li, B. Wang, and H. Liu, “Robot grasp detection using multimodal deep convolutional neural networks,” Advances in Mechanical Engineering, vol. 8, no. 9, p. 1687814016668077, 2016. [Online]. Available: https://doi.org/10.1177/1687814016668077

[12] H. Cao, G. Chen, Z. Li, J. Lin, and A. Knoll, “Lightweight convolutional neural network with gaussian-based grasping representation for robotic grasping detection,” arXiv preprint arXiv:2101.10226, 2021.

[13] Y. Jiang, S. Moseos, and A. Saxena, “Efficient grasping from rgb images: Learning using a new rectangle representation,” in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3304–3311.

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

[15] X. Yan, M. Khansari, J. Hsu, Y. Gong, Y. Bai, S. Pirk, and H. Lee, “Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks,” arXiv preprint arXiv:1906.08989, 2019.

[16] E. Johns, S. Leutenegger, and A. J. Davison, “Deep learning a grasp function for grasping under gripper pose uncertainty,” 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4461–4468, 2016.

[17] S. Levine, P. Pastor, A. Krizhevsky, 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, 03 2016.

[18] U. Viereck, A. ten Pas, K. Saenko, and R. W. Platt, “Learning a visuo-motor controller for real world robotic grasping using simulated depth images,” in CoRL, 2017.

[19] Z. Huang and J. Wang, “Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection,” Inf. Sci., vol. 522, pp. 241–258, 2020.

[20] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874–1883, 2016.

[21] D. Misra, “Mish: A Self Regularized Non-Monotonic Activation Function,” arXiv e-prints, p. arXiv:1908.08681, Aug. 2019.

[22] D. Kragic, H. Christensen, and F. A., “Survey on visual servoing for manipulation,” Computational Vision and Active Perception Laboratory, Fiskartorpsvägen, vol. 15, 02 2002.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference for Learning Representations, 2015.

[24] A. Depierre, E. Dellandréa, and L. Chen, “Jacquard: A large scale dataset for robotic grasp detection,” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3511–3516, 2018.

[25] U. Asif, J. Tang, and S. Harrer, “Grapnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices,” in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4875–4882. [Online]. Available: https://doi.org/10.24963/ijcai.2018/677

[26] ——, “Ensembler: Improving grasp detection using an ensemble of convolutional neural networks,” in British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018. BMVA Press, 2018, p. 10. [Online]. Available: http://bmva2018.org/contents/papers/0322.pdf

[27] S. Wang, X. Jiang, J. Zhao, X. Wang, W. Zhou, and Y. Liu, “Efficient fully convolutional neural network for generating pixel wise robotic grasps with high resolution images,” 2019 IEEE International Conference on Robotics and Biomimetics (ROBU), pp. 474–480, 2019.