2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Grasp Planning with CNN for Log-loading Forestry Machine

Elie Ayoub[1], Patrick Levesque[2] and Inna Sharf[3].

***Abstract*— Log loading constitutes a key operation in timber harvesting, and despite the recent spike of interest in introducing automation to the forestry sector, efficient and intelligent grasping of logs remains unresolved. This paper presents a grasp planning pipeline that relies on the identification of logs' characteristics and pose in the environment of a log-loading machine, to generate high quality grasps. The proposed pipeline involves replicating identified logs in a virtual environment where grasp planning is carried out by using a convolutional neural network and a virtual depth camera. The network relies solely on depth information and the virtual camera can be positioned at a strategically selected location or to follow a certain trajectory to enhance exposure of the logs, all this without having to move the log-loader's crane. The grasp planning pipeline is evaluated through simulated grasping trials and experiments on a large-scale log-loading test-bed with several configurations of wood logs ranging from a single to multiple logs. The grasp planning pipeline proved to be successful with a grasping rate of 98.33% in the simulated trials and 96.67% in the experimental trials. The grasp planner was able to overcome log characterization and localization uncertainties, thus allowing the log-loader to pick individual logs, and multiple logs at once when possible.**

I. Introduction

A. Background and Motivation

Automation in the forestry industry has been receiving increased attention over the past few years, as witnessed by the number of recent works that address robotics and AI related topics in the context of forestry operations. The lack of skilled machine operators, with many in the labour force close to retirement, coupled with low student participation in educational and training programs for forestry machine operators, make the shift from human-operated to automated forestry machines a pressing necessity [1], [2]. Efforts have been made to assist operators and to enhance human-machine cooperation through automating some decisions for the operator [3] or enabling teleoperation [4], but few works aim to achieve complete autonomy for the tasks of tree felling and log loading operations. For the log loading problem specifically, which is one of the key operations in delivering timber from forest to mill, majority of existing literature tackles the problem of path planning for the log-loader's crane [5], [6]. While few works attempt autonomous log grasping [6], [7], the results have always been demonstrated for single, isolated logs, rather than a diversity of possible configurations ranging from cluttered logs to large log piles. The absence of a robust grasp planning framework to produce grasps that allow single or multiple log grasping in complex log configurations motivates the work presented in this paper.

Fig. 1: FPInnovations Log-Loading Test-Bed

B. State of the Art

The inherent difficulties in robotic grasping have pushed recent research efforts in the direction of deep learning methods. Convolutional Neural Networks (CNNs), in particular, are a common choice for grasping solutions based on visual input [8], [9] due to their feature extraction abilities. Several state-of-the-art CNN multipurpose grasping approaches have been successfully used for accurate object picking and placing tasks [10]–[16]. For example, the approach in [10] creates a lightweight and fully CNN that relies on depth images only to generate a pixel-level grasp quality distribution on cluttered objects. The network's lightweight architecture allows for grasp predictions in real time, thus enabling grasping that follows targeted objects. The approach in [17] showcases the use of CNN from [11] to perform object segmentation and generate single-log grasp predictions of wood logs for log-loading applications, with a reported accuracy of 94.82%.

In the last several years, Reinforcement Learning (RL) methods have been intensely explored for general manipulation tasks involving robotic arms. The end-to-end learning capabilities of RL solutions allow them to map the state of the environment to robot actuator commands with minimal knowledge of their dynamics models [18]. For example, the approaches in [18], [19] show the possibility of using visual data from cameras for robotic grasping, without any knowledge of the robot arm's model. With respect to utilizing RL to

[1]Elie Ayoub is with the Department of Mechanical Engineering, McGill University, Montreal, QC H3A 2K7, Canada elie.ayoub@mail.mcgill.ca

[2]Patrick Levesque is a Software Developer with Software Electronic Applications, FPInnovations, Pointe-Claire, QC H9R 3J9, Canada Patrick.Levesque@fpinnovations.ca

[3]Inna Sharf is a Professor at the Department of Mechanical Engineering, McGill University, Montreal, QC H3A 2K7, Canada. She is currently a Lead Researcher at FPInnovations, Pointe-Claire, QC H9R 3J9, Canada inna.sharf@fpinnovations.ca

automate large-scale hydraulic machines, some works have considered excavator arms [20]–[23] and hydraulic forestry cranes [6], [24]. The authors of [6] investigate learning successful actuator-space control policies to grasp single logs with a forestry crane. The agent receives information about the pose of the target log and the end-to-end learned policy outputs the motor commands to navigate to the log and grasp it. This technique was implemented in simulation and resulted in 97% accuracy when the reward function didn't include energy consumption costs, and 93% when it did.

A different kind of learning approach to log grasping relies on a Generative Adversarial Network (GAN) to create graspability heatmaps in [25] that predict whether logs are graspable or not, based on RGB or RGB-D image inputs of clusters of logs. This approach provides potential log candidates to be grasped rather than a precise grasp to execute. Although most of the aforementioned approaches aim for automation of log-loading machines, some have explored other modalities, like teleoperation. The solution in [4] relies on structural light cameras to detect and localize logs in real time, and to place them in a virtual environment. The operator can then teleoperate the log-loader through feedback from said virtual environment to pick up the logs.

Several works attempt to tackle other log-grasping related issues, like detecting the presence of logs in the grapple. The work in [26] presents a grapple design with integrated proximity sensors capable of detecting the distance to logs around the grapple and the presence of logs between its tongs. This design is implemented in [7] on a down-scaled crane model and a CNN from [27] that predicts good grasp locations based on RGB-D images to augment the crane with a sense about the quality of grasps that it performs.

C. In this Paper

The main contributions of this paper are:

1) a modified CNN implementation to predict the grasp location and orientation. Training is done on a generic object dataset, but successfully applied to log grasping.
2) a novel grasp planning pipeline which uses a virtual camera in a virtual world created based on segmented log characteristics of real logs to run a CNN that generates the target grasp location. Using a virtual camera allows for quick grasp planning with minimal resources since the log loader does not have to move.
3) the proposed pipeline is capable of handling groups of logs, not just individual logs when predicting grasps.
4) experimental demonstration of the proposed grasp planner on a large-scale log-loading test-bed with multiple log configurations ranging from a single log to mini-piles of logs.

II. Grasp Planning Pipeline

A. Bird's Eye View

The proposed grasp planning pipeline takes a set of log parameters as input (pose, length, diameter) and outputs the location and orientation for a log-loader's grapple to move to where the grapple performs the grasping operation by closing its tongs. The work presented in this paper relies on segmentation to generate the aforementioned input parameters and a CNN to decide on the output grapple pose. To progress from input to output, a virtual environment created in Gazebo is employed as an intermediate step, where logs are recreated, post-processed, and positioned based on their input parameters. A virtual depth camera is then camera placy positioned in the virtual world at a point that provides good exposure of the logs to generate a target grasping position and orientation for the log-loader's grapple. A schematic of the pipeline is presented in Figure 2.

B. Log Segmentation

Since our grasp planner requires the pose and characteristics of individual logs in order to replicate them in the virtual environment, log detection and segmentation were selected as one of the more suitable methodologies to extract this information. The work in [28] introduces a log segmentation dataset named TimberSeg 1.0 that contains 220 images of 2500 segmented logs with their bounding boxes and mask annotations. The majority of images were collected through dash-cams that were placed on forwarder machines during their operation in forests over several months and through various weather conditions. Conveniently, the Mask2Former segmentation network from [29] was trained on the TimberSeg dataset and implemented by the authors of [28] on the FPInnovations log-loading test-bed employed for implementation and experimentation on our grasp planning pipeline. The implemented approach relies on a Swin-B transformer backbone that is pre-trained on the ImageNet-22k dataset [30] to extract features from RGB images and generate segmented masks. The Detectron2 library from Facebook is used to train the model on TimberSeg 1.0 and limit its application to the single-class detection task of identifying wood logs. The segmented log masks coupled with a point-cloud obtained from a depth camera are then combined to infer the length and diameter of the logs, and localize them in the scene around the log-loading test-bed.

C. Virtual Environment

Once the logs' poses and characteristics have been determined, virtual logs are instantiated in a blank Gazebo world as perfect cylinders with the corresponding poses, lengths, and diameters. The logs' lengths were reduced by the grapple's width and an extra 10% that was empirically found to avoid potential grasps at their edges, and to partially mitigate segmentation inaccuracies in length estimation. A random striped orange-like material is assigned to the cylinders for the sole purpose of visual clarity, since grasp planning relies exclusively on depth information. In the Gazebo world, the log-fixed frames coincide with their centroids and the world frame is set to represent the crane's base frame; thus, the positions and orientations of virtual logs in Gazebo match those of real logs with respect to machine's base frame.

As noted earlier, one of the unique features of our grasp planning scheme is that planning is carried out in the virtual (Gazebo) environment, with a virtual camera. To determine the placement of the camera in the Gazebo world, we proceed

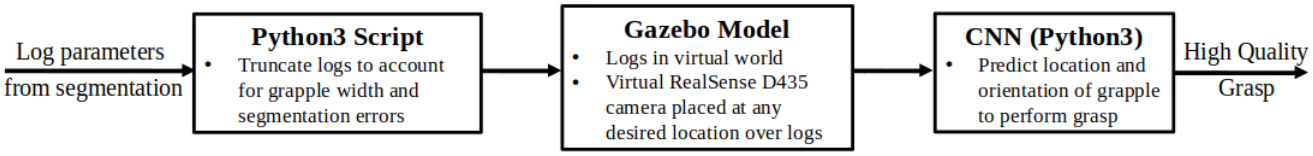


Fig. 2: Grasp planning pipeline

as follows. First, the grasping operational area is defined by computing the bounding box that contains the entirety of the generated virtual logs. This is done by determining the positions of the ends of each log $\mathbf{p}_i^{GW}$ ($i = 1, 2$ for the two ends) in Gazebo world frame using:

$$\mathbf{p}_{1,2}^{GW} = \mathbf{p}_L^{GW} + \mathbf{R}_L^{GW}\mathbf{p}_{1,2}^{L} \tag{1}$$

where $\mathbf{p}_i^L$ denotes the corresponding end positions in the log frame, $\mathbf{p}_L^{GW}$ is the position of the log-fixed frame and $\mathbf{R}_L^{GW}$ is the rotation matrix from the Gazebo world frame to a log-fixed frame.

Having computed all extremity locations, we locate the minimum and maximum position component ranges in the three directions, $\{x, y, z\}_{min/max}$, and form a cuboid with sides of length $(x_{max} - x_{min})$, $(y_{max} - y_{min})$, and $(z_{max} - z_{min})$. A virtual depth camera can now be created above the logs, at any desired location. Alternatively, it can swiftly follow a specified path while taking continuous depth shots to allow for grasp planning from multiple angles and averaging grasp predictions as new depth images are processed. The log-loader can then navigate to the target location immediately, once virtual grasp planning is completed. For the simulations and experiments presented in this paper, a RealSense D435 depth camera was instantiated at a static position $\mathbf{p}_C^{GW}$, 3 meters above the centroid of the cuboid defining the grasping operational area. This camera placement provides an uncut view of the logs from a reasonable distance.

D. Convolutional Neural Network

The final step of the grasp planning pipeline is performed through a CNN that takes as input a depth image of a configuration of logs, and computes a grasp map describing the predicted grasp quality and grapple pose at every pixel of the input depth image. We formulate our problem, similar to [10], as learning a function M to convert a depth image I to a grid-like grasp map $\tilde{G}$ of dimensions $\tilde{X} \times \tilde{Y}$ that predicts the grasp quality Q, and the grapple orientation angle $\tilde{\Theta}$ distributions over the pixels of a given depth image:

$$\tilde{G} = (Q, \tilde{\Theta}) = M(I) \in \mathbb{R}^{2 \times \tilde{X} \times \tilde{Y}} \tag{2}$$

A grasp instance from the grasp map provides the quality q and the grapple angle θ at a single pixel, and can be represented as:

$$\tilde{g} = (q, \tilde{p}, \tilde{\theta}) \tag{3}$$

The grasp quality q is a scalar quantity, $q \in [0, 1]$, quantifying the chances of grasp success, where a location with $q = 1$ is most likely to result in a successful grasp. Deciding on the optimal grasping location is often complex, as there are no general rules to determine grasps of high quality, with numerous possibilities [31], [32]. Therefore, to train the network, we rely on the grasping datasets that have been pre-labeled by humans to provide high quality grasp locations. Let $\tilde{p}$ denote the image coordinates $(\tilde{x}, \tilde{y})$ of the pixel and $\tilde{\theta}$ represent the grapple angle in the world frame. The grasping target $\tilde{g}^*$ is then defined by the pixel coordinates $(\tilde{x}^*, \tilde{y}^*)$ on the depth image that correspond to the highest pixel value of the grasp quality distribution. The target grapple angle is then the angle at the same point in the angle distribution, that is:

$$(\tilde{x}^*, \tilde{y}^*) = argmax_{(\tilde{x},\tilde{y})} Q$$
$$\tilde{g}^* = (q^*, \tilde{p}^*, \tilde{\theta}^*) = (Q_{(\tilde{x}^*,\tilde{y}^*)}, (\tilde{x}^*, \tilde{y}^*), \tilde{\Theta}_{(\tilde{x}^*,\tilde{y}^*)}) \tag{4}$$

The target grasp can then be converted from $\tilde{g}^*$ in the 2D pixel space into g^* in the log-loader's base frame (represented by the Gazebo world frame) by defining two transformations that transform from the grasp map's pixel space to the virtual camera's optical depth frame, and from the camera's optical depth frame to the Gazebo world frame.

1) Network Architecture: The chosen CNN architecture was inspired by the network presented in [10] because of its lightweight topology, however, with the addition, subtraction and manipulation of some layers based on a series of trials where the number of convolutional layers, type of operations, number, size of filters, and other network attributes were sequentially varied to achieve the highest test accuracy rate and a faster than in [10] prediction rate. The network's average prediction time is a critical consideration when deciding on its architecture, since fast predictions executable at every Gazebo frame will be necessary if the virtual camera is to follow a path and send continuous depth information, rather than take one static depth image as is done in the present implementation. The optimal resulting network possesses a fully convolutional architecture, where the input depth image of size 300x300 is first scaled down to a 100x100 grid through average pooling, before undergoing any convolution. The scaled down image is subjected to seven intermediary convolutional layers, one Max-pooling operation and two 2D bilinear upsampling operations to be re-scaled to the original 300x300 input size. Therefore, the network requires 78 803 parameters and its architecture is summarized in Table I.

2) Training on Cornell Grasping Dataset: A labeled grasping dataset is required to train our neural network to produce grasp predictions for arbitrary log configurations. For this work, we choose to use one of the available standardized datasets, the Cornell Grasping dataset[1]: it contains RGB-D images of 885 real objects with multiple grasp labels

[1]http://pr.cs.cornell.edu/grasping/rect_data/data.php — As of September 12, 2022, the website is down.

TABLE I: Convolutional Neural Network Summary

Layer	Output shape	Number of parameters
Input	(1, 300, 300)	0
AvgPool2D	(1, 100, 100)	0
Conv2D	(16, 100, 100)	1952
Conv2D	(16, 100, 100)	12560
ReLU	(16, 100, 100)	0
BatchNorm2D	(16, 100, 100)	32
MaxPool2D	(16, 50, 50)	0
Conv2D	(16, 50, 50)	6416
ReLU	(16, 50, 50)	0
Conv2D	(16, 50, 50)	6416
BatchNorm2D	(16, 50, 50)	32
Conv2D	(32, 50, 50)	12832
Conv2D	(32, 50, 50)	12832
ReLU	(32, 50, 50)	0
BatchNorm2D	(32, 50, 50)	64
UpsamplingBilinear2D	(32, 100, 100)	0
Conv2D	(16, 100, 100)	12816
ReLU	(16, 100, 100)	0
UpsamplingBilinear2D	(16, 300, 300)	0
Conv2D	(1, 300, 300)	17
Conv2D	(1, 300, 300)	17
Conv2D	(1, 300, 300)	17

per object denoted by rectangles. Following the data pre-processing and training framework in [10], the dataset was augmented 10 times with random zooms, crops, and rotations to obtain 8850 images. The 640x480 images from the dataset were cropped at their center to form 300x300 images which provide the input to the CNN. The true labels that were used were generated by setting the grasp quality inside the Cornell grasp rectangles to 1 and 0 elsewhere. The grasping angles were based off the orientations of the Cornell rectangles and were decomposed into their respective $\cos 2\theta$ and $\sin 2\theta$ components, as seen in [10]. This decomposition allows for the generation of unique angles in the range of $[-\pi/2, \pi/2]$ that can be used by the log-loader's symmetric grapple. The grapple opening information can be inferred from the rectangles in the Cornell dataset, as seen in [10]. However, this was not implemented here, since the log-loading test-bed used for experimentation does not allow for controlling the opening of the grapple beyond fully open or closed.

The network training was performed by minimizing the total mean squared error denoted by:

$$MSE_{total} = MSE_Q + MSE_{\cos 2\tilde{\Theta}} + MSE_{\sin 2\tilde{\Theta}}$$
$$= \frac{1}{NN_p}\sum_{n=1}^{N}\sum_{i=1}^{N_p}[(\hat{Q}_i - Q_i)_n^2 + (cos(2\hat{\tilde{\Theta}}_i) - cos(2\tilde{\Theta}_i))_n^2$$
$$+(sin(2\hat{\tilde{\Theta}}_i) - sin(2\tilde{\Theta}_i))_n^2] \quad (5)$$

where variables denoted by a hat represent the true labeled data, N represents the number of training examples and N_p represents the number of pixels per image which is 90 000. To test the accuracy of the network before employing it in our grasp planning pipeline, 10-fold cross validation was applied on the augmented Cornell dataset and resulted in an 83.8% accuracy. A grasp's success was judged by expressing the resulting prediction as a rectangle similar to the Cornell dataset label. If the overlap area between true and predicted rectangles normalized by the area of their union was greater than 25%, and their orientations were within 30° of each other, the grasping prediction was considered to be correct. This method is commonly used for grasp evaluation in [10], [13]–[16] which also tackle grasp prediction and synthesis.

III. Description of Log-Loading Test-bed

A. Crane

The experiments and simulations presented in this paper were conducted on the log-loading (the crane) test-bed at FPInnovations (see Figure 1).This facility was built up from the originally purchased commercial system PALMS 4.71[2]. The arm, referred to as the crane of the test-bed, is integrated on a fixed-base platform which houses the hydraulic system to drive the crane and a bunk to hold the logs. The crane itself has the topology typical of a log-loading machine, such as the forwarder: it is a seven degree-of-freedom under-actuated arm with the first four joints (RRRP) to position the tip of the boom, followed by two passive rotary joints to support the grapple (C4 model from PALMS) and the last joint (the rotator) to orient the grapple with respect to the logs. The grapple tongs are actuated open-loop to fully open or close the grapple. The crane has a maximum reach of 7.1 m.

To enable closed-loop joint control, the actuated revolute joints have been instrumented with absolute joint encoders (Magres–EAM360-B-CANopen[3]) measuring the output (link) angle. The extension of the fourth (prismatic) joint is measured with a magnetic sensor/band combination (MSA501/MBA501).[4] In addition, the crane is instrumented with a ZED 2i stereo camera[5] rigidly mounted on a stick link (see Figure 1); the camera provides the image data for log segmentation, as discussed in the following section.

The control hardware enabling the closed-loop joint control of the crane includes a laptop communicating through Ethernet with PLC, which in turn generates the commands to the hydraulic valves of the crane using a PD control law. The information from the ZED camera is received by the laptop where the vision-based grasping pipeline is evaluated. The hardware and software architectures are illustrated in Figure 3 and 4 respectively. ROS Noetic was used as middleware for communication between the software components.

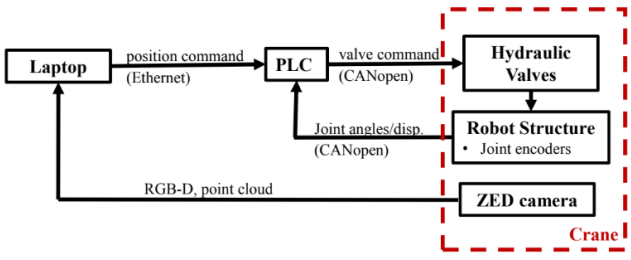


Fig. 3: Hardware Architecture

[2]https://www.palms.eu/forest-cranes?productID=58

[3]https://www.baumer.com/us/en/product-overview/rotary-encoders-angle-sensors/industrial-encoders-absolute/36-mm-integrated/eam360-b/eam360-b—canopen/p/27928

[4]https://www.siko-global.com/en-ca/products/magline-magnetic-linear-and-angular-measurement/magnetic-sensors/msa501

[5]https://store.stereolabs.com/products/zed-2i

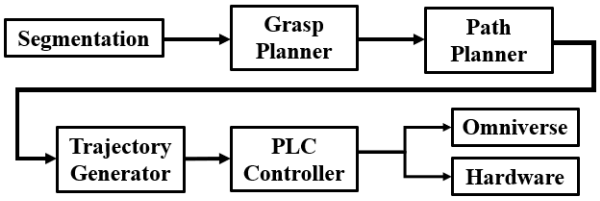


Fig. 4: Software Architecture

B. Environment

The crane test-bed is located in a fenced outdoor parking area of FPInnovations. The set of logs employed for the grasping experiments comprised around a dozen logs of dry red pine, with lengths in the range of 2.5-2.8 m and diameters in the range of 0.1-0.3 m. The logs were placed on asphalted ground, in different configurations as described in Section IV.

IV. Results

A. Test Scenarios

To quantify the success of our grasping pipeline, we define 12 test configurations to replicate common arrangements of logs that a log-loader might encounter during its operation (see Table II). These configurations present opportunities for grasping a single log in some cases and multiple logs (crossed logs and mini-piles) in other cases. The configurations were reviewed by a forestry researcher[6] and confirmed as representative of routine log-picking operations.

For every configuration, 5 grasping attempts are performed for a total of 60 grasping tests. A single test includes execution of the grasp planning pipeline introduced in Section II, the crane repositioning to the grasping target location, the grapple commanded to perform the grasp, and finally, the crane raising the grasped log(s) till they cleared the ground. For each of the five attempts per configuration, the logs were arranged to maintain the configuration, but placed at random locations and orientations. The positions of the logs were also cycled between each other during same-configuration attempts (a log that was on top of a pile during an attempt may be at the bottom of the pile on the next attempt). The lengths and diameters of the logs that were used for different configurations were chosen at random. A grasping attempt on configuration 9 can be seen in Figure 5.

B. System Modeling in NVIDIA Omniverse Isaac Sim

A model of the test-bed was created using Isaac Sim from the NVIDIA Omniverse platform. Starting with the 2D CAD models of the crane components, 3D models of the parts were generated, meshed and saved in STL format. The URDF and STL files were then imported into Omniverse to create the virtual test-bed. The masses of the main components of the crane were obtained from the manufacturer and the other inertial parameters estimated based on the mass and geometry information. Hydraulics of the test-bed are not modelled. A camera with matching properties to the ZED 2i was positioned in the latter's place and configured to relay RGB-D data through an Isaac Sim built-in ROS bridge.

[6]Peter Hamilton reviewed all configurations. He is a Senior Researcher at FPInnovations, Pointe-Claire, QC H9R 3J9, Canada

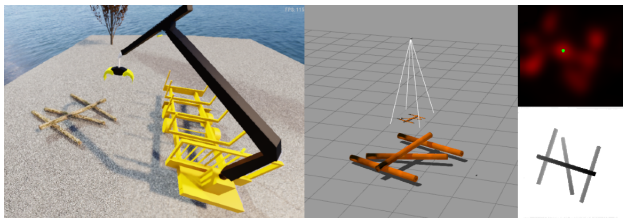


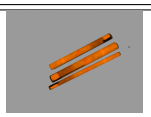
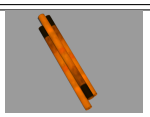
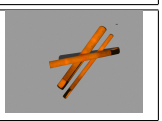
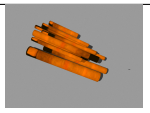
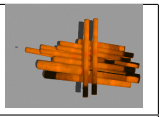
Fig. 5: Configuration 9 in Isaac Sim (left), post-segmentation in Gazebo (middle), corresponding depth image from Gazebo virtual camera (lower right), grasp quality distribution with optimal grasping point marked by green dot (upper right).

C. Simulated Results

For each grasping attempt, grasp success is achieved if the grapple wraps its tongs around at least one log, and the grasped log(s) are raised off the ground at the end of the test. Successful grasp attempts are categorized into optimal and sub-optimal grasps. Optimal grasps make the most sense for a certain configuration and are likely to be chosen by an operator, such as, grasps of the top log in configurations 3 and 9, or grasps at the geometric center of single and parallel logs in configurations 1 and 2. In addition, the grapple angle is aligned with the direction of logs to be grasped during an optimal grasp. Sub-optimal grasps are considered in three categories: Non Intuitive (NI) for grasps of logs that are under other logs and grasps at the edges of single or parallel logs, Segmentation Error (SE) for grasps that were not optimal because of segmentation inaccuracies, such as, logs missing or inaccurate log localization, and finally, Misaligned Angle (MA) for grasps with grapple angles that did not align well with the direction of the targeted logs.

Table II presents the number of successful simulated grasp attempts out of 5 for every configuration and their optimal/sub-optimal categorization. Out of the 60 performed attempts, the log-loader succeeded at grasping and lifting a log or a bunch of logs during 59 (sum of successful grasps in table II) tries (98.33% success). Multiple log grasping was consistently observed for configurations that allowed it: all logs in configurations 2, 4, 5, 6, 7, and more than 1 log in configurations 10, 11, and 12. Out of the 59 successful attempts, 48 were optimal and 11 were sub-optimal, the latter including 4 NI and 7 SE cases. Segmentation errors were expected to be dominant in the simulated trials since the segmentation network was trained on images of real wood logs. The virtual wood material and the resolution of the simulation's image also played a role in deteriorating the segmentation quality. Observed segmentation errors included logs shifted from their real positions, logs completely missed and unsegmented, individual logs that were segmented as multiple logs because part of them was obstructed, and logs with widely inaccurate lengths or diameters. Note that segmentation inaccuracies occurred frequently in the simulated trials but were only reported when they caused sub-optimal grasps. Indeed, the grasp planner was able to produce optimal grasps even with multiple segmentation problems. For the only failed grasp attempt for configuration 5, the combination of a non intuitive grasp location at the edge of the logs and a

TABLE II: Simulated Grasp Outcome for Test Log Configurations

Log Config.	1	2	3	4	5	6
Picture						
Grasp Success	5	5	5	5	4	5
↳Optimal	5	3	4	5	3	4
↳Sub-Optimal	n/a	SE, NI	NI	n/a	SE	NI
Log Config.	**7**	**8**	**9**	**10**	**11**	**12**
Picture						
Grasp Success	5	5	5	5	5	5
↳Optimal	3	5	4	2	5	5
↳Sub-Optimal	SE, SE	n/a	NI	SE, SE, SE	n/a	n/a

segmentation error in the form of a shift in the logs' positions caused the grapple to completely miss its target.

D. Experimental Results

Experiments were carried out on the crane test-bed over two days (August 17th and 24th, 2022). Both days were a mix of sunny and cloudy conditions, providing a wide range of lighting conditions for the camera. Weather and visibility conditions have the potential to influence log segmentation results only since the remainder of the pipeline relies on the replicated logs in Gazebo. The same 12 configurations were employed and the same result reporting procedure from the simulated results was followed for the experimental trials. Table III presents the number of successful experimental grasp attempts out of 5 for every configuration and their optimal/sub-optimal categorization. Out of 60 attempts, the log-loader successfully grasped and lifted a log or a bunch of logs in 58 (sum of successful grasps in table III) tries (96.67% success). As in the simulated trials, multiple log grasping was consistently observed throughout configurations that allowed it: all logs in configurations 2, 4, 5, 6, 7, and multiple logs in configurations 10, 11, and 12. Of the 58 successful attempts, 51 were optimal while the remaining 7 were sub-optimal, the latter consisting of 4 NI, 1 SE, and 2 MA categories. Although some segmentation and localization errors were present in the experimental trials, they were less frequent than in the simulated trials. Testing on real hardware, however, presented new sources of errors, such as inaccuracies in the positioning of the grapple due to flexibility of the crane (estimated to be of the order of 0.1 m). The point-cloud generated by the ZED 2i camera was noisy enough to cause the segmentation and log-localization to be consistently slightly in error. Nonetheless, the grasp planner was able to circumvent these errors during most grasping attempts. The two grasp failures were due to segmentation missing a large portion of the logs and generating abnormally large diameters. This caused the reconstruction of log pieces with a width larger than their length in Gazebo, which in turn resulted in predicted grapple angles to be nearly 90° off the correct angles, ultimately leading to failed grasp attempts.

TABLE III: Experimental Grasp Outcomes

Log Config.	1	2	3	4	5	6
Grasp Success	5	5	5	5	5	5
↳Optimal	5	5	4	3	3	5
↳Sub-Optimal	n/a	n/a	NI	NI, NI	MA, SE	n/a
Log Config.	**7**	**8**	**9**	**10**	**11**	**12**
Grasp Success	5	5	5	4	4	5
↳Optimal	5	5	4	4	4	4
↳Sub-Optimal	n/a	n/a	NI	n/a	n/a	MA

V. Conclusion

This work introduced a robust grasp planning pipeline to enable intelligent grasping of various configurations of logs (individual logs and piles). Logs were identified and localized using segmentation, and recreated as virtual cylinders in Gazebo. A virtual camera then relayed depth information to a CNN that predicted the best grasping location on a single or multiple log configuration and the grapple angle to perform the said grasp. The simulated and experimental trials that were used to validate the pipeline resulted in similar high grasping success rates (98.33% and 96.67% respectively) and confirmed the ability of the presented grasp planner to generate accurate grasps that efficiently target multiple logs, and overcome uncertainties and errors that occur in determining the log characteristics and localizing them in the scene. The presented work is the first to tackle log grasping autonomy beyond grasping a single log at a time.

Future work will include training the CNN on a Gazebo logs dataset that we collected and labeled with the aid of a forestry researcher. We aim to study the effect of training with a dataset that is specific to our task on the grasping accuracy of our pipeline and on the number of optimal vs. sub-optimal grasps that are generated. Future plans also include clustering logs and planning the grasp order of clusters when tens of logs piled in multiple configurations and locations are situated in the vicinity of the log-loader.

Acknowledgments

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN). The log-loading test-bed was developed and is supported by FPInnovations.

REFERENCES

[1] F. Huq and I. Branch, "Skills shortages in canada's forest sector," *Canadian Forest Service*, 2007.

[2] S. Kollarova, "Innovation and advanced technology use in the canadian forest sector," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2014.

[3] E. Yousefi, D. P. Losey, and I. Sharf, "Assisting operators of articulated machinery with optimal planning and goal inference," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2832–2838.

[4] S. Westerberg and A. Shiriaev, "Virtual environment-based teleoperation of forestry machines: Designing future interaction methods," *Journal of Human-Robot Interaction*, vol. 2, no. 3, pp. 84–110, 2013.

[5] D. Ortiz Morales, S. Westerberg, P. X. La Hera, U. Mettin, L. Freidovich, and A. S. Shiriaev, "Increasing the level of automation in the forestry logging process with crane trajectory planning and control," *Journal of Field Robotics*, vol. 31, no. 3, pp. 343–363, 2014.

[6] J. Andersson, K. Bodin, D. Lindmark, M. Servin, and E. Wallin, "Reinforcement learning control of a forestry crane manipulator," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2121–2126.

[7] S. Weiss, S. Ainetter, F. Arneitz, D. A. Perez, R. Dhakate, F. Fraundorfer, H. Gietler, W. Gubensäk, M. M. D. R. Ferreira, C. Stetco *et al.*, "Automated log ordering through robotic grasper."

[8] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.

[9] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, and X. Shen, "Robotics dexterous grasping: The methods based on point cloud and deep learning," *Frontiers in Neurorobotics*, vol. 15, p. 73, 2021.

[10] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International journal of robotics research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[11] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 452–13 458.

[12] W. Hu, C. Wang, F. Liu, X. Peng, P. Sun, and J. Tan, "A grasps-generation-and-selection convolutional neural network for a digital twin of intelligent robotic grasping," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102371, 2022.

[13] L. Chen, P. Huang, Y. Li, and Z. Meng, "Edge-dependent efficient grasp rectangle search in robotic grasp detection," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 2922–2931, 2020.

[14] 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.

[15] Y. Yu, Z. Cao, Z. Liu, W. Geng, J. Yu, and W. Zhang, "A two-stream cnn with simultaneous detection and segmentation for robotic grasping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[16] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection. in 2018 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3511–3516.

[17] S. Ainetter, C. Böhm, R. Dhakate, S. Weiss, and F. Fraundorfer, "Depth-aware object segmentation and grasp detection for robotic picking tasks," *arXiv preprint arXiv:2111.11114*, 2021.

[18] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.

[19] S. Joshi, S. Kumra, and F. Sahin, "Robotic grasping using deep reinforcement learning. arxiv," 2020.

[20] P. Egli and M. Hutter, "Towards rl-based hydraulic excavator automation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2692–2697.

[21] ——, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[22] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9778–9785, 2022.

[23] Q. Lu, Y. Zhu, and L. Zhang, "Excavation reinforcement learning using geometric representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4472–4479, 2022.

[24] R. Dhakate, C. Brommer, C. Böhm, H. Gietler, S. Weiss, and J. Steinbrener, "Autonomous control of redundant hydraulic manipulator using reinforcement learning with action feedback."

[25] J. Andersson, "Predicting gripability heatmaps using conditional gans," 2022.

[26] L.-M. Faller, C. Stetco, and H. Zangl, "Design of a novel gripper system with 3d-and inkjet-printed multimodal sensors for automated grasping of a forestry robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5620–5627.

[27] 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.

[28] J.-M. Fortin, O. Gamache, V. Grondin, F. Pomerleau, and P. Giguère, "Instance segmentation for autonomous log grasping in forestry operations," *arXiv preprint arXiv:2203.01902*, 2022.

[29] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[31] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1824–1829.

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