Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 2210–2219

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Evaluation of Grasps in an automatic Intermodal Container Unloading System

Aleksei Kharitonov[a,*], Alice Kirchheim[a], Maximilian Hentsch[b], Johannes Seibold[a], Wolfgang Echelmeyer[c]

[a]*Aalen University, Beethovenstr. 1, 73430 Aalen, Germany*
[b]*Steinbeis Innovation gGmbH, Adornostr. 8, 70599 Stuttgart, Germany*
[c]*Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, Germany*

Abstract

Today, many industrial tasks are not automated and still require human intervention. One of these tasks is the unloading of oversea containers. After the end of transportation to the sorting center, the containers must be unloaded manually for further sending the parcels to the recipients. A robot-based automatic unloading of containers was therefore researched. However, the promising results of the system developed in these projects could not be commercialized due to problems with its reliability. Mechanical, algorithmic or other limitations are possible causes of the observed errors. To analyze errors, it is necessary to evaluate the results of the robot's work without complicating the existing system by adding new sensors to it. This paper presents a reference system based on machine learning to evaluate the robotics grasps of parcels. It analyzes two states of the container: before and after picking up one box. The states are represented as a point cloud received from a laser scanner. The proposed system evaluates the success of transferring a box from an overseas container to the sorting line by supervised learning using convolutional neural networks (CNN) and manual labeling of the data. The process of obtaining a working model using a hyperband model search with a maximum classification error of 3.9 % is also described.

Keywords: automatic parcel unloading evaluation; laser scanner; machine learning point cloud; automated container unloading;

1. Motivation

Today, the automation of many logistical processes is state of the art. This includes processes such as the transport of load carriers with automated guided vehicles (AGVs), the sorting of packages in distribution centers, and the storage of piece goods in warehouses. However, processes with only partially defined basic conditions cannot be automated economically today. This includes the handling of varying piece goods with different properties (size, shape, weight), that are only partially visible and accessible inside a load carrier. In particular, unloading of swap bodies and containers

* Corresponding author.

1877-0509
10.1016/j.procs.2021.08.234

is a challenging process [4]. This problem is becoming more and more important in the context of the ever-increasing growth of global trade and the resulting increase in global parcel volumes. But the automation of unloading processes from containers is still unsolved nowadays. The problem of automating the process of unloading goods from shipping containers has been an active research field for many years, as Stoyanov et al. show [25]. But despite the increasing importance of the problem, it has still not been solved today.

Previous approaches include an attempt to develop a fully automated system for unloading parcels [21, 8]. However, the promising results of such systems could not be commercialized due to problems with their reliability. A high error rate while unloading goods (e.g. dropped parcels) required frequent human intervention. The cause of the errors has not yet been determined. One or more possible causes exist, such as incorrect calculation of trajectories, mechanical constraints such as a reduced degree of freedom, mechanical inability to correctly grasp packages, or inaccurate package detection. An in-depth analysis of the systems is therefore necessary to determine and differentiate the causes of the errors. To correctly analyze and fix these flaws and to distinguish between them, an evaluation of the grasps of the unloading system is needed beforehand. The challenge of determining the success of a robotic grasping operation can be approached in several ways. One of the possibilities is a technical modification of the system and the installation of additional sensors. But this approach leads to increasing complexity of the unloading system and additional costs. Another way is to create a set of static rules that evaluate the result of grasps based on the scanning data of the container after every single grasp of the parcel unload robotics. But to write rules using human knowledge for every possible erroneous grasp is very inefficient. The paper presents a reference implementation for an automated detection of erroneous grasp processes of the unloading system based on laser scanner data using machine learning approach, in particular based on convolutional neural networks (CNN). In this work, we show that with the help of machine learning methods and the data of the laser scanner, an automated evaluation of gripping processes is possible. Because the point cloud is the most widely used data format for representations of the ranges in 3D robotics, we focus on that format of the input data. The machine learning model generation for the evaluation of grasps using hyperband search method is described [11].

To evaluate the success of the unloading process, it is necessary to classify the changes that have occurred in the sensor data before and after the grasp. The paper presents a reference implementation for an automated detection of erroneous grasp processes of the unloading system based on laser scanner data using machine learning.

Automated robotic grasps and the state of the art for automated unloading grasps and the related machine learning applications will be discussed in section 2. The problem statement, materials and methods are presented in section 3. The main part consists of the presentation of the developed procedure and the evaluation of the results that are presented in subsection 4.3 and section 5. In the end, the work is concluded in section 6.

2. Previous research

Automated unloading grasps. The topic of automated robot grasps of heterogeneous objects has been researched for many years [3]. It is still a challenging problem to extract objects with different shapes using robotics nowadays [22, 23]. Many technical difficulties are caused due to impossibility to acquire a full 3D image to come to conclusion about an extracting strategy for a given object. Even with modern sensors such as laser scanners or RGB-D cameras only a reconstruction of the object's front surface can be obtained [22]. Beyond that, sensors must be suitable for industrial use and be as inexpensive as possible. Automated container unloading of a cluttered scene is being researched by Vaskevicius et al.[29]. But there are still no error-free systems known yet.

Machine learning for robotic grasps. Achievements in the research of machine learning allow a wide range of applications on different technical tasks, including automation of robotics grasps. There are different works that address the automation of robotics grasps using machine learning methods. The research progress of machine learning on robotic grasping is discussed by different research groups, in particular by Li et al. and by Caldera et al. [12, 5]. Mahler et al. proposed datasets and algorithms to train machine learning-based methods to plan robot grasps [15]. Redmon et al. designed a system based on a CNN to predict robotic grasps of objects in RGB-D images [20]. Liang et al. have researched a problem of localizing robot grasp configurations directly from the point cloud for different object shapes using YCB object and model dataset [13, 6]. This approach is based on the PointNet architecture that operates with the point cloud data format and classifies N points in the Euclidean 3D space into k classes without voxelization of the

input space and with no use of collections of 2D images generated from 3D space [18, 34, 27]. Thus, it drastically reduces computational complexity of the neural networks as well as the throughput. However, there are some advances, like adaptive sampling, which shows a better model performance [31]. Zhang et al. presented a system based on a CNN that evaluates the success of the robotics grasps based on the RGB images of the robot arm with up to 92.8% accuracy and the inference time of a single image of 524 ms [33]. However, this approach is based only on the RGB data for the evaluation of the robotic grasps.

As 3D scanner data is heavily used in robotic applications, machine learning models have to operate with such data [1]. A common technique is to use a *volumetric representation* of a model using voxels which is followed by 3D convolutions to explore the discriminative features of the objects [19, 16]. Another approach is the *multi-view* method, which is based on *dimension reduction*, e.g. to generate 2D images of the scene from different virtual camera views [26]. In this case, for n images, n CNN models are trained and a view-pooling is applied on the whole feature space afterwards. Other methods are *point cloud*-based [18]. They try to overcome the huge computation complexity of voxel convolution and show higher speeds in segmentation and classification tasks [32]. Another approach is *geometry-based* analysis. A geometry-based method was researched by Tatarchenko et al [28]. Their approach is based on tangent convolutions and operates directly on surface geometry [28]. Another geometry-based way with dimension reduction was proposed by Lin et al. [14]. They introduced a convolutional operator that projects a 3D patch onto a 2D grid plane with a use of 2D convolution afterwards.

3. Objective and methodology

Objective. Through examination of 3D scanner data before and after the unloading of an object from a container, that represent its states before and after the robotics grasp, the goal is to determine, whether the occurred grasp between these states was successful or not.

Materials and methods. At first, the processing of the 3D scanner data is explained. It is based on a single-view 2D representation of the 3D scenes, which are captured by a laser scanner. In the next step, a gradient image is generated, which is the difference between the projection images, see subsection 4.1. The idea behind the gradient image generation is: The success of a grasp can be interpreted based on the gradient image, i.e. in the best case a single region with an area, that contains high gradient values, resembles a parcel, that has to be classified as a successful grasp (parcel moved). In the easiest case of an unsuccessful unloading grasp, two gradient areas have to be distinguishable in the gradient image, in which one area has to be marked by mostly prevalent negative gradient values. If that is the case, a parcel was removed from the scene, thus the distances in the area, where the parcel was, have to increase. Another area has to be distinctive by the positive gradient values, i.e. a fallen parcel has to decrease the distances to the measured points before the parcel appeared on the scene. The experiment is based on the data captured from a real robotics system for automatic container unloading, which is described in section 4.2. In the last step, a hyperband model search is conducted to obtain an optimal classification model for the described problem, see subsection 4.3 [11].

4. Data preprocessing and machine learning model development

4.1. Single-view 2D representation of the 3D scene

In the following, we describe creation of the training set. Two given point clouds from the laser scanner represent the states of the container before and after the unloading of one parcel. To estimate the success of the grasp process, it is necessary to evaluate the difference between two images. The created image is called a gradient image in the following discussion. Because the 3D scans can be captured with different scan sample rates, the number of points in the point clouds can differ, thus, it is necessary to scale the images to a common resolution. This is achieved by interpolating their pixel values to a static size. A human eye can capture the difference and evaluate the robotics grasp without the need to see the the point cloud from multiple angles. A single front view on the container is enough to make the decision about the success of the unloading process. Therefore, it has to be feasible to transform the 3D view to a single-view 2D representation and obtain sufficient evaluation performance. In the single-view 2D image,

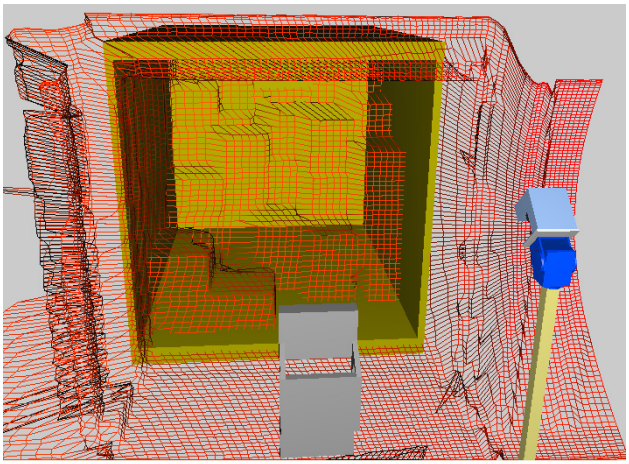


Fig. 1: The automatic unloading system overview. Loaded container (middle) contains parcels that have to be automatically ejected by the gripping system. A 3D laser scanner (right bottom) acquires the distances to the scan points. The scanned points are depicted in form of the mesh grid. The gripper (bottom middle) grabs parcels and moves them on a conveyor belt for further sorting.

each pixel represents the radial distance r to the point at a certain polar angle θ and at a certain azimuthal angle φ in a spherical coordinate system, where the position of the laser scanner is defined by $(0, 0, 0)$. To create a 2D image a field of view grid has to be defined as:

$$D = \begin{bmatrix} (\theta_{\min}, \varphi_{max}) & (\theta_1, \varphi_{max}) & \dots & (\theta_{\max}, \varphi_{max}) \\ (\theta_{\min}, \varphi_1) & (\theta_1, \varphi_1) & \dots & (\theta_{\max}, \varphi_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\theta_{\min}, \varphi_{min}) & (\theta_1, \varphi_{min}) & \dots & (\theta_{\max}, \varphi_{min}) \end{bmatrix}$$

It contains a pair of azimuthal θ and polar φ angles. Thus, the field of view is limited by $\varphi_{min}, \varphi_{max}, \theta_{min}$ and θ_{max}. We define a function $d(\theta, \varphi)$, where $d : \mathbb{R}^2 \to \mathbb{R}$ is the function returning the distance from the scanner to the reflection point on the remote surface for the given θ and φ, in which $\theta \in [\theta_{min}, \theta_{max}]$ and $\varphi \in [\varphi_{min}, \varphi_{max}]$, otherwise $r = \emptyset$.

The value for a point at $(r_i, \theta_i, \varphi_i)$ is calculated using nearest neighbor interpolation. In other words, for a given sample of existing 3D points $\{p_1, p_2, \dots, p_n\}$ at locations $\{(r_1, \theta_1, \varphi_1), (r_2, \theta_2, \varphi_2), \dots, (r_n, \theta_n, \varphi_n)\}$, to estimate the value p_i at some new point $(r_i, \theta_i, \varphi_i)$ an index j has to be found. The index j determined as $j = \arg\min \left| p_j - p \right|$, the value of p is then defined as $(r_j, \theta_i, \varphi_i)$. The 2D single-view image is then calculated as follows: For pixels $I_{ij} \in I$ of the image I, calculate value as $I_{ij} = d(D_{ij})$. To determine the difference between two images $I_1 \in \mathbb{R}^2$ and $I_2 \in \mathbb{R}^2$, a gradient $G \in \mathbb{R}^2$ can be computed as $G = \text{grad}(I_1, I_2)$ [2]. As a simplest gradient function we used an element-wise matrix subtraction: $G = I_2 - I_1$. The gradient information is used further as the input data for the machine learning model.

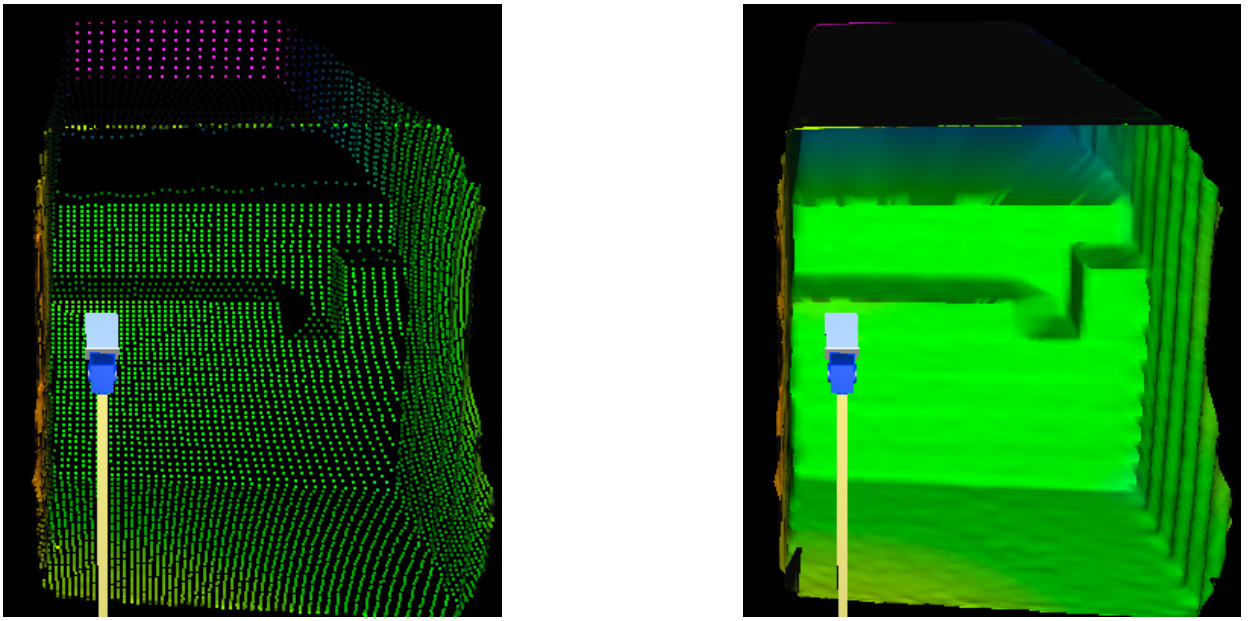
4.2. Experimental setup

Automated unloading system for overseas containers. The investigated system is shown in Figure 1. This parcel unloading system is developed for automated parcel unloading up to 70 kg from overseas containers [21]. The laser scanner creates a 3D point cloud that depicts the physical container state. After that, decision algorithms calculate the best object to grasp and a trajectory for the gripper. The gripper unloads a parcel from the container and puts it on the conveyor belt for further handling.

Dataset description. The following describes the dataset that serves as the basis for this work. The dataset was collected over 291 days of normal workload. Each state of the container was captured via a laser scanner [24]. The

seq_num	n_rows	distances/mm	yaws/°	pitches/°
85	98	[1770.0, 1751.0....]	[109.0, 108.0..., 30]	[-9.992795, -9.992795...]
86	112	[1775.0, 1747.0....]	[109.0, 108.0..., 30]	[-9.996149, -9.996149...]
...	...	...	...	...
79	123	[1414.0, 1876.0....]	[109.0, 108.0..., 30]	[-17.99441, -17.99441...]

Table 1: Dataset snippet. Each sequence represents a container state. The sequence number `seq_num` incrementally increases by one to distinguish the consecutive distribution of data. Number of rows `n_rows` defines how many point lines were acquired in a given sequence. Distances to points are saved as an array in a column. The number of elements inside the array of distances is defined as $(\max\{\text{yaws}\} - \min\{\text{yaws}\} + 1) \cdot \text{n\_rows}_i$.



(a) 3D view of measuring points of a loaded container.

(b) View of a loaded container after surface creation using meshing

Fig. 2: Container view visualized using laser scanner.

used laser scanner is a SICK LMS 200 from the manufacturer SICK AG. Each container scan was saved into a separate text file. A total of 193,971 files were collected. Each text file contains meta-information about field of view of the sensor, number of scanned rows and measured distances to the reflection points in a spherical coordinate system (r, θ, φ - radial distance, polar angle and azimuthal angle accordingly) and also their representation in an Euclidean coordinate system (x, y, z - coordinates). A sample of a text file using the acquired points is depicted on Figure 2a. A representation of the laser scanner and a container after surface creation for better visualization is shown on Figure 2b. Example data is shown in Table 1.

4.3. Development of the deep learning model

The development process of the DNN is depicted on Figure 3. The first step is to collect container images captured by the sensors for some period of time. Those will be used for the model training via machine learning techniques.

Data filtering. A dataset collected for many unloading sequences can contain damaged data, images captured for sensor calibration, distorted images, etc.. Thus, data filtering is necessary to get a better classification performance of the end model. Unusable images should be removed from the dataset or be repaired. For example, images that are taken in uncommon situations, like during the maintenance works or during the calibration, should be removed.

Training and test dataset creation. Our approach is based on supervised learning, which means, that the model will be trained with the ground truth information (label) that will be used during the training process. In other words, for every given input x_i, a label y_i should be provided, so that the trained function f will output $f(x_i) = y_i$. In our case, the output of the function is a discrete value, so the following is valid:

$$f(x_i) = \begin{cases} 0, & \text{if the grasp process } i \text{ was successful} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

Criteria for labeling. To evaluate the grasp of the robot, criteria have to be defined to distinguish between unsuccessful and successful unloading processes. It is labeled as successful, if all criteria are fulfilled. They are relative to S_i (container state before) and to S_{i+1} (container state after the single unload event). The following criteria describe a successful grasp process.

- S_i has one or more packets optically visible in the field of view,
- it can be optically interpreted that one or more packets in S_{i+1}, are out of the field of view.

If it was visually evident that only the robotics have changed their position, e.g. moved inside the container, but didn't grasp any parcel, then the pair of states is not considered to be a subject of evaluation and, therefore, was not labeled. In all other cases, the grasp of the robot is interpreted as unsuccessful.

To label the input data, a random unloading sequence from a total of 555 was chosen and the image pairs in the chosen sequence were labeled. From about 74,000 available images, around 4,000 pairs of the images were manually labeled and used for training and test of the model. After that, a gradient was computed between the pair of images by applying the gradient function G. The resulting sets of gradient images containing labels were divided into a training set and a test set. The training set contains 70 % of the manually labelled gradient images, while the test set contains the remaining 30 %.

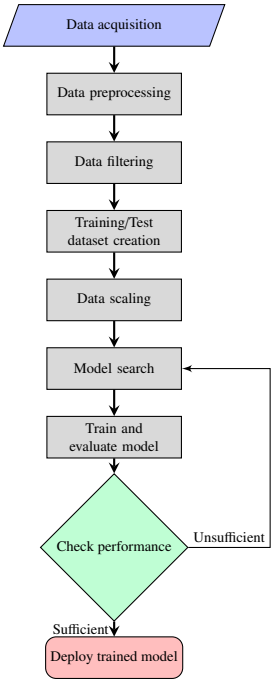


Fig. 3: ML model development

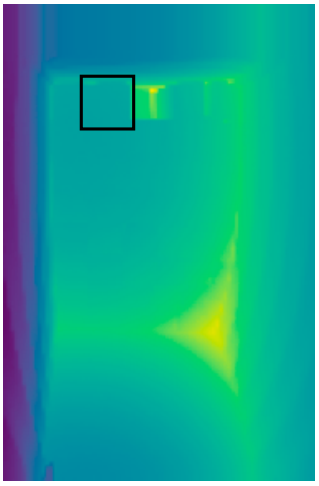
Data scaling. Data scaling can be useful to avoid exploding and vanishing gradients during the training. In our case, the minimal distance r_{min} was around 80 mm and the maximal distance in the image was $r_{max} = 8191$ mm. This shows a high variance of the input data. Thereby, a normalization or standardization can be applied to increase model stability during the training and to increase its sensitivity. Standardization was applied to scale the input data. This step has improved the overall performance of the classifier by around 5 %. The standardized value r_s is calculated using the formula $r_s = \frac{r_i - \bar{r}}{\sigma}$, where r_i is the original value, $\bar{r}$ is the mean value $\forall r$ and σ is the standard deviation of the dataset.

Model search. There is a great variety of machine learning model architectures that can be chosen. Searching for the optimal model architecture can be very time consuming. To automate the searching process of the best performing model, a hyperband search was carried out using the Keras Tuner library [17]. Because CNNs architecture performs highly accurate for image classification problems, an application of this kind of machine learning model architecture was a reasonable choice [9, 10, 30]. For the feature extraction, convolution layers were used. The number of layers was defined to be between 2 and 5. Kernel size was statically defined to be 3×3 with padding to the input size. The number of filters varied between 32 and 256 with a step of 32. Following each convolution, a batch normalization and Rectified Linear Unit (ReLU) activation function were applied. After every convolutional layer, either the maximum, the average or no pooling layers were tested for applicability. The last layer was flattened and attached to one fully connected layer. The number of neurons varied between 10 and 1000 with a step of 10. The activation function was

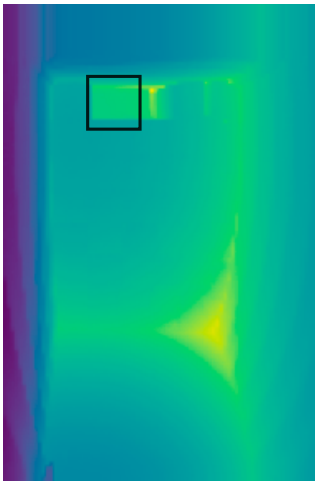
statically defined to be ReLU. The dropout layer is applied with a rate between 0 and 0.5 with a step of 0.1. The last layer contains one output neuron that uses the sigmoid activation function. The optimizer of the model was set to Adam with default parameters. The loss function `binary_crossentropy` was used for the model.

Train and evaluate model. After finding the best performing model, it was trained with the prepared training dataset and tested with the test dataset. The training and test were carried out using the Keras framework [7].

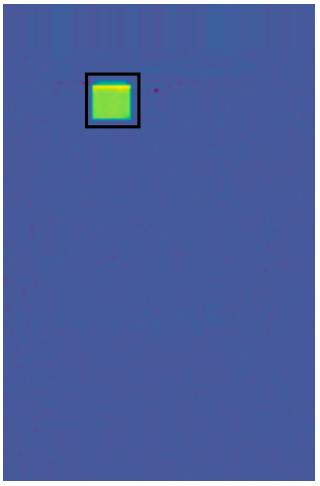
Performance check. To make a decision about usability of the model, metrics are necessary to be defined. The criteria for the model performance were chosen based on a tolerable failure rate of false negatives under 5 % and false positives under 1 %. This means, than the detection of the successful unloading operations is more important that the detection of the unsuccessful ones.



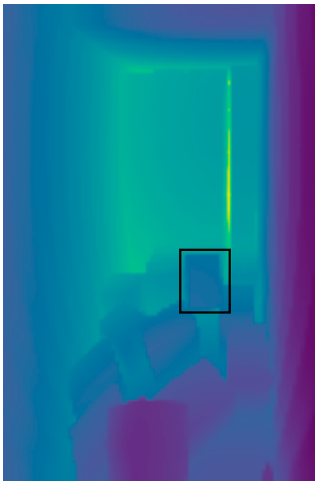
(a) Container view before a successful unloading event



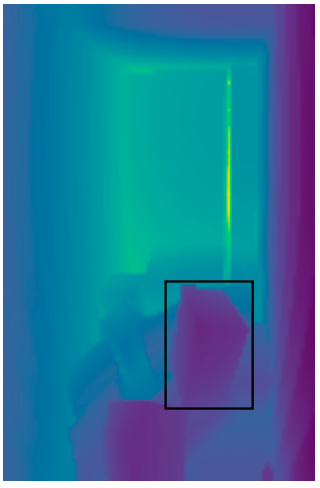
(b) Container view after a successful unloading event



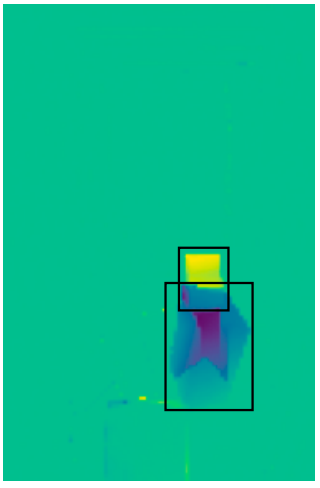
(c) Difference between the states before and after the unloading event



(d) Container view before an unsuccessful unloading event



(e) Container view after an unsuccessful unloading event



(f) Difference between the states before and after the unloading event

Fig. 4: Container range view visualized using laser scanner. A successful grasp of a box is represented on 4a, 4b and 4c. On the image 4d, a packet was grasped and moved towards the conveyor belt, but it fell down after being moved by the gripper. At the next captured container state, which is depicted in 4e, the parcel appears on the image randomly stacked on top another box. The difference between two images contains the original box position colored yellow and its new position colored dark blue. On the difference image represented in 4f, the original and the new position of the grasped box are partially superimposed. By looking at the difference image with human eye, it is difficult to say, whether the unloading was successful or not. The trained deep learning model was able to recognize such cases and evaluate the grasp correctly.

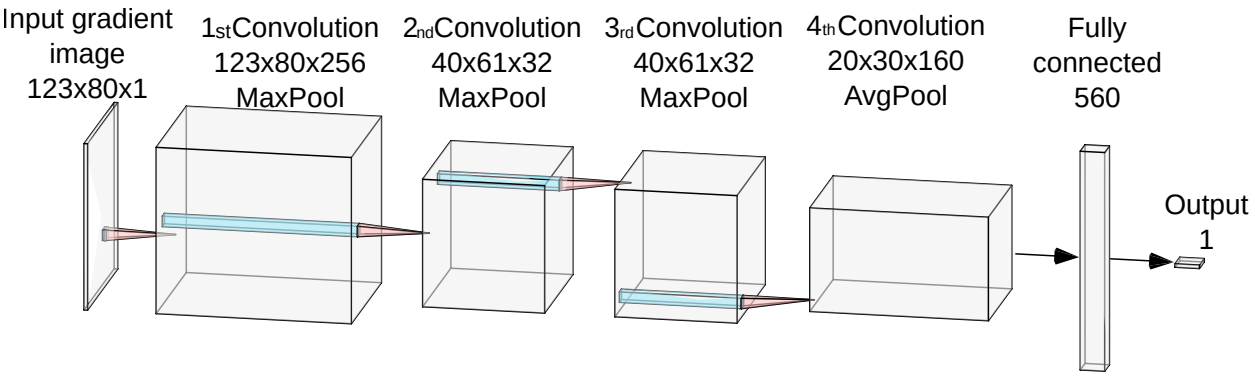
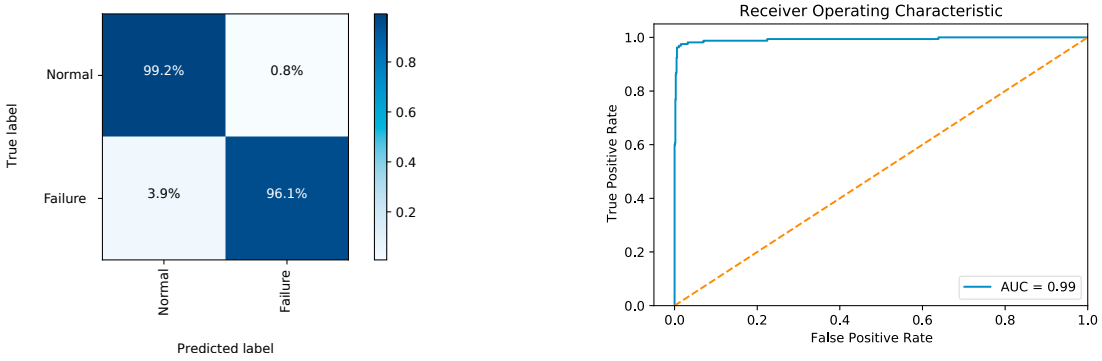


Fig. 5: Architecture of the best trained model. The output is the evaluation result: unloading successful yes/no.



(a) Confusion matrix of the classifier

(b) Receiver operating characteristics of the classifier

Fig. 6: Achieved performance of the classifier

5. Results and discussion

During the hyperparameter search, 11,155 different models were trained. The structure of the model with the best performance is shown in Figure 5. It consists of around 2.1 million parameters. The performance of the proposed classification approach has achieved an accuracy of 99.2% for unloading operations, that have been successfully carried out. The classification of the failed grasps was achieved with 96.1%, see Figure 6a. The area under the receiver operating characteristics curve (ROC AUC) for the classifier is 0.99, which is shown in Figure 6b.

A single inference time for one difference image was about 250 milliseconds, carried out on a Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz. This is a negligible amount of time, compared to the time of several seconds of a single grasp. This means, that the proposed algorithm is applicable for a real-time use during the machine operation, without a significant increase of the unloading time of a container. As shown by the metrics, classification of the successful grasps is an easier task than the classification of the unsuccessful grasps, because fewer artifacts are present on the difference images, see Figure 4c, which results in a better convergence of the loss function. Classification of the unsuccessful grasps is a harder task, because the difference image contains a much higher amount of different shapes of objects, such as moved neighbor parcels and fallen boxes. The trained model shows a false negative error rate of 3.9 %, which defines, how many unsuccessful grasps were incorrectly classified, in contrast to 0.8 % of the false positive error rate, that is related to incorrect classification of the successful grasps.

As the analysis of the error rate shows, most of the false negatives (around 41 %) were produced by incorrect classification of the cases, where the robot couldn't grasp the parcel correctly, slightly shifted it and has moved back. About 14 % of all false negatives occurred after the robotic arm was captured on one of both images of the container states. From the total of false positive cases, i.e. if the model classifies the image as a failure grasp, but it was a good

carried out grasp, ca. 26 % of the states had a shifted neighboring parcel. Around 37 % of the false positives had strong artifacts in the image, which were caused due to re-reflections from metallic constructions in the field of view of the scanner.

Suggestions for improvements in future works. In the current work, an approach using model search was researched. One of the possible approaches is to use transfer learning of existing DNNs. This could show an improved performance of the model and better convergence. To reduce the error rate, improved data preprocessing may be required, e.g. elimination of the artifacts, that occurred due to re-reflections. Another approach may be successful by using a region-based classification. This method does not use the whole image, but compares only a region of interest in some surrounding area of the highest gradient areas in the image. It may produce higher efficiency and a more compact model due to reduction of the input data and, therefore, produces smaller amount of weights needed for the model and less computational complexity.

6. Conclusion

The proposed method based on sequence difference analysis between the images of the overseas container states using a CNN has shown its applicability for the evaluation of grasp processes during the automated unloading. This approach uses data from already built-in laser scanners and doesn't require additional sensors that have to be retrofitted in the existing system. In addition to that, this is an important step towards automated analysis of grasps algorithms, evaluation of mechanical actions, object detection and a movement towards self-learning robotics for automated container unloading. Based on this work, a reinforcement learning approach could be developed, where the robot control system and its algorithms have to learn from mistakes, that were made in the past, via received feedback from the proposed evaluation system.

Acknowledgements

The project was funded by Stiftung Kessler & Co. für Bildung und Kultur under the title "EXPLOR – Autonomisierung in der Intralogistik".

Author contribution statement

A.Kh. wrote the manuscript, performed the laboratory work and experiments, designed the concept and implemented the software in consultation with J.S.. A.Ki. conceived the project, supervised the laboratory work and participated in writing the motivation and directed the structure of the manuscript. M.H. provided useful ideas for the work, participated in structuring and improvement of the manuscript. W.E. provided the dataset and the documentation for the robotics. Percentage contributions are A.Kh: 75 %, A.Ki: 10 %, M.H.: 5 %, J.S.: 5 %, W.E.:5 %. All authors read and approved the final manuscript.

References

[1] Ahmed, E., Saint, A., Shabayek, A.E.R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., Ottersten, B., 2019. A survey on deep learning advances on different 3d data representations. arXiv:1808.01462.

[2] Avanaki, A.N., 2009. Exact global histogram specification optimized for structural similarity. Optical Review 16, 613–621. doi:10.1007/s10043-009-0119-z.

[3] Bicchi, A., Kumar, V., 2000. Robotic grasping and contact: a review, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), pp. 348–353 vol.1. doi:10.1109/ROBOT.2000.844081.

[4] Bonini, M., Kirchheim, A., Echelmeyer, W., 2012. Challenges in the application of autonomous cognitive systems within logistics, in: Autonomous Robot Systems and Competitions, pp. 33–38.

[5] Caldera, S., Rassau, A., Chai, D., 2018. Review of deep learning methods in robotic grasp detection. Multimodal Technologies and Interaction 2. URL: https://www.mdpi.com/2414-4088/2/3/57. online; accessed 04 June 2021.

[6] Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M., 2015. The ycb object and model set: Towards common benchmarks for manipulation research, in: 2015 International Conference on Advanced Robotics (ICAR), pp. 510–517. doi:10.1109/ICAR.2015.7251504.

[7] Chollet, F., et al., 2015. Keras. https://keras.io. Online; accessed 04 June 2021.

[8] Echelmeyer, W., Bonini, M., Rohde, M., 2014. From Manufacturing to Logistics: Development of a Kinematic for Autonomous Unloading of Containers. Advanced Materials Research 903, 245–251. doi:10.4028/www.scientific.net/AMR.903.245.

[9] He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. arXiv:1512.03385.

[10] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

[11] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research 18, 1–52. URL: http://jmlr.org/papers/v18/16-558.html. online; accessed 04 June 2021.

[12] Li, Y., Lei, Q., Cheng, C., Zhang, G., Wang, W., Xu, Z., 2019. A review: machine learning on robotic grasping, in: Verikas, A., Nikolaev, D.P., Radeva, P., Zhou, J. (Eds.), Eleventh International Conference on Machine Vision (ICMV 2018), International Society for Optics and Photonics. SPIE. pp. 775 – 783. doi:10.1117/12.2522945.

[13] Liang, H., Ma, X., Li, S., Görner, M., Tang, S., Fang, B., Sun, F., Zhang, J., 2019. Pointnetgpd: Detecting grasp configurations from point sets, in: 2019 International Conference on Robotics and Automation (ICRA), pp. 3629–3635. doi:10.1109/ICRA.2019.8794435.

[14] Lin, Y., Yan, Z., Huang, H., Du, D., Liu, L., Cui, S., Han, X., 2020. Fpconv: Learning local flattening for point convolution. arXiv:2002.10701.

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

[16] Muzahid, A.A.M., Wan, W., Hou, L., 2020. A new volumetric cnn for 3d object classification based on joint multiscale feature and subvolume supervised learning approaches. Computational Intelligence and Neuroscience 2020, 5851465. doi:10.1155/2020/5851465.

[17] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al., 2019. Keras Tuner. https://github.com/keras-team/keras-tuner. Online; accessed 04 June 2021.

[18] Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv:1612.00593.

[19] Qi, C.R., Su, H., Niessner, M., Dai, A., Yan, M., Guibas, L.J., 2016. Volumetric and multi-view cnns for object classification on 3d data. arXiv:1604.03265.

[20] Redmon, J., Angelova, A., 2015. Real-time grasp detection using convolutional neural networks, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1316–1322. doi:10.1109/ICRA.2015.7139361.

[21] RobLog, . Roblog project (robotic logistics). URL: http://52367068.fn.freenet-hosting.de/. online; accessed 30 March 2021.

[22] Saxena, A., Driemeyer, J., Ng, A.Y., 2008. Robotic grasping of novel objects using vision. The International Journal of Robotics Research 27, 157–173. doi:10.1177/0278364907087172.

[23] Shimoga, K., 1996. Robot grasp synthesis algorithms: A survey. The International Journal of Robotics Research 15, 230–266. doi:10.1177/027836499601500302.

[24] Stoyanov, T., Mojtahedzadeh, R., Andreasson, H., Lilienthal, A.J., 2013. Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications. Robotics and Autonomous Systems 61, 1094 – 1105. URL: http://www.sciencedirect.com/science/article/pii/S0921889012001431, doi:https://doi.org/10.1016/j.robot.2012.08.011. selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011).

[25] Stoyanov, T., Vaskevicius, N., Mueller, C.A., Fromm, T., Krug, R., Tincani, V., Mojtahedzadeh, R., Kunaschk, S., Mortensen Ernits, R., Canelhas, D.R., Bonilla, M., Schwertfeger, S., Bonini, M., Halfar, H., Pathak, K., Rohde, M., Fantoni, G., Bicchi, A., Birk, A., Lilienthal, A.J., Echelmeyer, W., 2016. No more heavy lifting: Robotic solutions to the container unloading problem. IEEE Robotics Automation Magazine 23, 94–106. doi:10.1109/MRA.2016.2535098.

[26] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015a. Multi-view convolutional neural networks for 3d shape recognition. arXiv:1505.00880.

[27] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G., 2015b. Multi-view convolutional neural networks for 3d shape recognition, in: Proc. ICCV.

[28] Tatarchenko, M., Park, J., Koltun, V., Zhou, Q.Y., 2018. Tangent convolutions for dense prediction in 3d. arXiv:1807.02443.

[29] Vaskevicius, N., Mueller, C.A., Bonilla, M., Tincani, V., Stoyanov, T., Fantoni, G., Pathak, K., Lilienthal, A., Bicchi, A., Birk, A., 2014. Object recognition and localization for robust grasping with a dexterous gripper in the context of container unloading, in: 2014 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1270–1277. doi:10.1109/CoASE.2014.6899490.

[30] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. arXiv:1611.05431.

[31] Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S., 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. arXiv:2003.00492.

[32] Yang, D., Gao, W., 2020. Pointmanifold: Using manifold learning for point cloud classification. arXiv:2010.07215.

[33] Zhang, H., Lan, X., Zhou, X., Wang, J., Zheng, N., 2017. Vision-based robotic grasp success determination with convolutional neural network, in: 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 31–36. doi:10.1109/CYBER.2017.8446360.

[34] Zhou, Y., Tuzel, O., 2017. Voxelnet: End-to-end learning for point cloud based 3d object detection. arXiv:1711.06396.