CHALMERS
UNIVERSITY OF TECHNOLOGY

Optical Load Detection

Load Weighing for Construction Machines using Stereo Vision and Convolutional Neural Networks

Master's thesis in Systems, Control and Mechatronics

DANIEL STRÅHLE
KEVIN WINGÅRD OLSSON

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022:21

Optical Load Detection

Load Weighing for Construction Machines using Stereo Vision and Convolutional Neural Networks

DANIEL STRÅHLE
KEVIN WINGÅRD OLSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Gothenburg, Sweden 2022

Optical Load Detection
Load Weighing for Construction Machines using Stereo Vision
and Convolutional Neural Networks
Daniel Stråhle
Kevin Wingård Olsson

Supervisor: Mathias Andreasson, CPAC Systems AB
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2022:21
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Demonstration of depth measurement using the stereo camera mounted on an excavator.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Optical Load Detection
Load Weighing for Construction Machines using Stereo Vision and Convolutional Neural Networks
Daniel Stråhle
Kevin Wingård Olsson
Department of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Accurate excavation monitoring is important for the handling of materials within the construction industry. Modern construction machines provide built-in systems for weighing handled goods. In this thesis, an alternative optical weighing system is developed and implemented for an excavator and a wheel loader. The optical system detects and provides the volume and weight of the handled material through fill-factor estimation. The methodology is based on depth data and images captured by a stereo camera, mounted on the machines. By using a region-based convolutional neural network (CNN), localization of material and fill-factor estimation are managed jointly. Material classification is also proved to be possible using gathered images and a simple CNN. By combining the fill-factor and information about the material, weight is obtained. Evaluations reveal that the system measures fill-factor to mean absolute percentage errors (MAPE), relative to the maximum capacity of the excavator and the wheel loader, of 3.3 % and 3.0 % respectively.

Keywords: Excavation Monitoring, CNN, Faster R-CNN, RPN, Range Sensor, Stereo Camera, Computer Vision, Material Classification.

Acknowledgements

This project would not be possible without all the external support we have received. First of all, we want to thank our great supervisor Mathias Andreasson, who introduced us to the project, the office, and CPAC Systems AB. His joyful ambitions and knowledge contributed to the large inspiration for advancing the project. Many thanks are dedicated to our examiner Peter Forsberg, who contributed to the project with insightful discussions and advice. Furthermore, the achieved results would not be possible without Niklas Sjöstedt, who supported us with knowledge regarding the machines and operating them during the data acquisitions. Additional thanks to Marcus Carlsson for operating the wheel loader during one of the acquisition campaigns. At last, we share great gratitude to everyone at CPAC Systems AB for welcoming us to the office and everything around it, as well as for supporting us with the project.

Daniel Stråhle and Kevin Wingård Olsson, Gothenburg, June 2022

Thesis advisor: Mathias Andreasson, CPAC Systems AB
Thesis examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

List of Acronyms

API application programming interface

CNN convolutional neural network

FPS frames per second

IMU inertial measurement unit
IOU intersection over union

LIDAR light detection and ranging

MAE mean absolute error
MAPE mean absolute percentage error

R-CNN region based convolutional neural network
RADAR radio detecting and ranging
RGB red, green and blue
ROI region of interest
RPN region proposal network

STD standard deviation
SVM support-vector machine

TOF time-of-flight

Contents

List of Figures

List of Tables

1

Introduction

In this chapter, the background and related work encompassing optical load detection are covered. Furthermore, the objective and scope of the project are presented.

1.1 Background

Excavation and material transportation are major works within the construction industry. For large-scale construction industries, there is a great endeavor of improving safety and productivity to reduce costs, work labor and environmental impact. Hence, there is a desire to increase the efficiency of material loading and transportation. This can be accomplished by high-accuracy sensing and excavation progress monitoring. For proper handling and transportation of materials, it is important to know the type and the weight of the goods. The load weighing systems available for excavators and wheel loaders today are mostly built-in systems that measure axis loads. However, this method has a cumbersome setup. Hence, alternative or complementary methods are of interest. One idea is an optical load detection system to automatically detect the type, volume and weight of the goods. As of now, not many attempts have been made of optical weight estimation for construction machines. The development of such a system has the potential to open up many new possibilities. For example, retrofitting machines with internal sensors used by current methods might not be possible. In such a case, an external optical system may be the sole option.

1.2 Related Work

An optical system commonly involves object detection and classification, which are both common tasks within computer vision and machine learning domains. 3D reconstruction from 2D images and the concept of estimating the volume of objects using computer vision are not new either. There have been attempts at volume estimation using optical sensors for various applications. To name a few, Rundgren uses a multi-view system for 3D reconstruction and volume estimation of timber loads [1]. Another work by Artaso and López-Nicolás uses a time-of-flight (TOF) camera and two structured light cameras for measuring the volume of merchandise in a logistics application [2]. Another example is dietary assessment. The number

of calories is estimated from food images by classifying and estimating the volume of food [3].

More related to construction and excavation monitoring, previous work by Lu et al. uses a neural network-based approach for fill-factor estimation and bucket detection on construction machines [4]. Fill-factor refers to the ratio of the loaded material's volume with respect to the maximal load capacity of the bucket or container. According to the authors, bucket fill-factor estimation remains one of the key challenges in the automation of construction machines. Lu et al. identify some of the difficulties in estimating the volume of handled material. A particular difficulty is that on-board weighing systems are unable to estimate loaded volume without external information about the density of the material. However, an optical solution would be able to solve the issue.

The method proposed by Lu et al. comprises three stages: pre-processing, machine learning and post-processing [4]. In the pre-processing stage, depth images are generated from data captured by a stereo camera. The depth images are used for predicting bucket fill-factor through machine learning. Lastly, the predictions go through post-processing based on a probabilistic approach.

Another previous work is conducted by Rasul et al. [5]. They use integrative methodologies for effective excavation progress monitoring. The work involves two aspects: volume estimation and 5D mapping. For volume estimation, two methods are used: direct estimation and indirect estimation. For the direct estimation, the excavated ground volume is estimated by comparing a 3D measurement of the ground with a reference measurement. Both measurements are captured by a stereo camera. An issue with the direct estimation is the risk of an occlusion area in the digging space, which hinders accurate ground detections. The indirect estimation uses the bucket volume, given by a 3D model, as a reference together with a measurement of a filled bucket. The difference between the 3D model and measurement is the excavated volume.

The second aspect investigated by Rasul et al. is the idea of 5D mapping for material classification [5]. It includes information on the excavated ground in terms of geometric space and material properties using information from several sensors. 3D data obtained by the stereo camera is fused with intensity data, obtained by a light detection and ranging (LIDAR) sensor, and ground resistive force, obtained through pressure data of the excavator. By fusing information, more accurate material classifications can be achieved.

1.3 Purpose and Goals

This project aims at improving excavation progress monitoring. A comprehensive solution for detecting, classifying and estimating the volume and weight of loaded materials using optical means is to be developed. By accurately monitoring the loaded material, trucks can be filled optimally and the movement of materials can

be closely tracked.

The purpose of the project is to develop a load detection system. The system should work both as a standalone system, and as a complement to existing weighing and classification systems. In this project, the targeted construction equipment are excavators and wheel loaders.

The project composes of several parts, which can be summarized by the following objectives:

- Choice and implementation of hardware
- Identify the container of material in sensor data
- Material volume through fill-factor estimation
- Material classification
- Material weight estimation

The main objectives are the fill-factor estimation of the container and material classification. For this application, the container corresponds to the buckets of the construction machines. When investigating and working on the objectives, the following questions are answered:

1. How can an optical system be implemented for volume and weight estimation of handled material, for a construction machine?
2. What performance, in terms of mean absolute percentage error (MAPE), can be achieved for fill-factor and weight estimation of construction material using convolutional neural networks?
3. What performance, in terms of classification accuracy, can be achieved for material classification using convolutional neural networks?

1.4 Limitations

The system developed in the project should be a comprehensive solution in the terms of solving fill-factor, volume and weight estimations of loaded materials, as well as material classification. The focus is not to provide a scalable and generalized solution, but rather to provide a basis containing all the components which can be further developed and improved. Hence, the scope of the project is limited to:

- Consider only one excavator and wheel loader model.
- Consider only one bucket size and type for each machine.
- Requirements for the operator are specified such as how to level and orientate

the buckets when the data is gathered.

- There are no strict requirements of efficiency in terms of the computational speed of the system.
- Data processing and predictions are performed offline.

2

Theory

This chapter explains the necessary theory for understanding the method. The idea is to use depth information. Therefore, techniques for acquiring this information are presented, with a focus on binocular vision used in stereo cameras. Lastly, the system utilizes machine learning, in specific convolutional neural networks and the Faster R-CNN framework.

2.1 Range Sensors

Range sensors refer to sensors that capture 3D information from the sensor's viewpoint [6]. The depth is usually measured as the distance to the closest object(s), either as a single point, scanning plane or as a whole image containing depth values for each point. The acquired data is useful for many applications, such as robotics and automation. Perception about the world is required for navigation or for determining 3D properties of objects. There are plenty of sensors available today to measure depth, some of which are presented here.

Several sensors revolve around two principles: active triangulation and time-of-flight (TOF) [7]. Active triangulation generally refers to structured light depth cameras. A projector is used to illuminate a surface with a known pattern, which is then acquired by a camera. The acquired image contains a superimposed version of the pattern. Depth can be obtained by comparing the patterns and performing geometrical reconstruction. The performance of a structured light setup depends on the choice of patterns, algorithms and the scene [8].

The other family of range sensors revolves around TOF. They emit a signal, and receive the signal sometime later by reflection. By using the time between transmitting and receiving the signal, as well as the speed of it, the distance to the reflection point can be calculated [9]. This concept has been used in radio detecting and ranging (RADAR) technology for a long time, using electromagnetic waves [7]. LIDAR is another TOF sensor that uses laser [8]. It usually has a large field-of-view at the cost of providing sparse depth information [10].

2.2 Stereo Vision

A range sensor based on binocular vision is the stereo camera. It utilizes two or more cameras combined with triangulation for estimating depth. The following sections introduce the theory behind computer vision and binocular vision.

2.2.1 3D Perception

Cameras map the 3D world to 2D images using projections [11]. A consequence of projecting the world to a surface is the loss of one dimension, namely depth. The projection is irreversible, which means that the depth dimension can not be recovered from a single 2D image. However, the knowledge of 3D data is essential for applications dealing with robot vision, automatic navigation, automotive safety and many others [8]. Hence, acquiring 3D properties from environments has been an active subject of research for a long time. Humans and most animals use a binocular visual system (two eyes) [12]. The key point of a binocular vision system is the difference between the left and right images due to slightly different perspectives. This mechanism inspired image-based 3D reconstruction using images captured from multiple points of view.

2.2.2 Binocular Vision

Figure 2.1 depicts an overview of a binocular vision system, with essential axes and distances to be able to calculate the distance to a point P, with world coordinates (x, y, z). The distance between the sensor and a point in the scene is also known as depth.

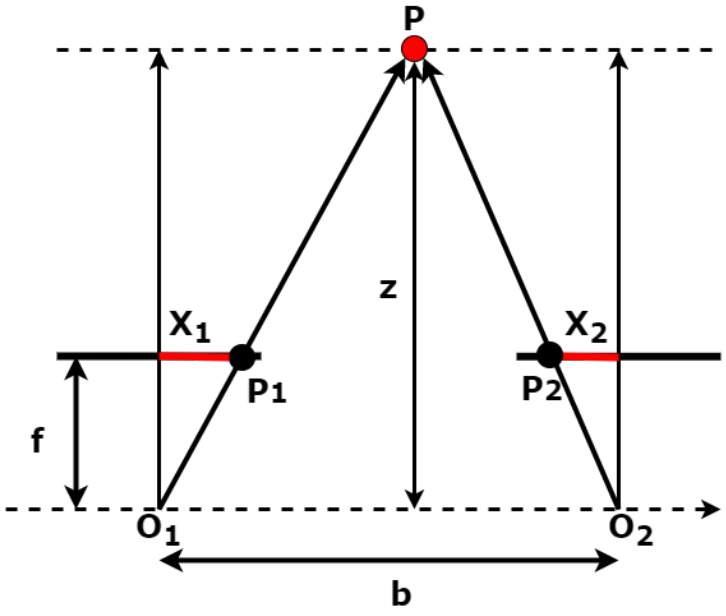


Figure 2.1: Triangulation scheme of stereo vision with relevant geometry for estimating the distance, z, to a point P with world coordinates (x, y, z). Each image plane has its own coordinate system.

In Figure 2.1, O_1 and O_2 represents the optical centres of the cameras. The optical distance between the cameras is called the baseline, denoted with b. Point P is projected through the image plane of the left and right cameras at points P_1 and P_2 with local image coordinates (X_1, Y_1) and (X_2, Y_2) respectively. Focal length, denoted f, is the distance between the optical center of one camera to its image plane. The aim is to find the distance z to the point P through triangulation. Two triangles are formed: PO_1O_2 and PP_1P_2. By utilizing uniformity, the distance is described by Equation (2.1).

$$\frac{b}{z} = \frac{b - X_1 + X_2}{z - f} \tag{2.1}$$

Extracting the distance z from Equation (2.1) yields Equation (2.2).

$$z = \frac{bf}{X_1 - X_2} = \frac{bf}{d} \tag{2.2}$$

In Equation (2.2), $d = X_1 - X_2$ denotes the disparity. It represents the horizontal difference between corresponding pixels in the left and right images. The disparity is an essential component of the binocular vision system. Since the rest of the values in Equation (2.2) are constant and usually known, the task becomes to obtain the disparity for each pixel in the image pair to retrieve the distance z. The collection of disparity values describing the whole scene is called a disparity map. The procedure of obtaining the disparity map is a matter of matching corresponding pixels in each image using various matching algorithms.

2.2.3 Stereo Matching Methods

Stereo matching is a vital part of 3D reconstruction since it is the main step to retrieve depth information. However, stereo matching is challenging due to difficulties such as noise, specular surfaces, ambiguous regions, repetitive patterns, transparent objects and occlusions [13].

The methods for finding point correspondences can be broadly divided into two categories: sparse and dense methods [12]. Sparse methods are usually feature-based, meaning that sets of potential image locations are extracted and then matched between the images. Nevertheless, sparse features can be difficult to recover in a 3D scene. Dense methods are correlation-based and produce disparity estimates for all image regions [14]. The dense stereo correspondence method thus produces a dense disparity map with a disparity estimate for each pixel.

Scharstein and Szeliski propose a taxonomy for stereo algorithms in [14], where most perform the following steps to solve the correspondence problem:

1. Matching cost computation
2. Cost (support) aggregation
3. Disparity computation/optimization
4. Disparity refinement

The procedure depends on the specific algorithm used. Matching cost computation compares pixels in the left and right images using a cost function. A lower cost means two pixels are more likely to be a matching pair. The cost function is usually based on the difference in pixel intensities. However, the cost calculated from two pixels might not have sufficient information to determine the match. Thus, cost aggregation can be used to include information from nearby points. A frequent method is to use a predefined window of some size for averaging or summing costs from a region. The size of the window heavily affects the result and thus has to be chosen wisely.

The disparity computation can be divided into local and global methods. Local methods emphasize the matching cost computation and the cost aggregation steps. The final disparities are chosen as the disparity for each pixel with the minimum cost value [15]. However, the local approach is sensitive to image noise, occlusions and blur areas. On the other hand, the global approach uses a more intelligent disparity decision strategy by including assumptions about the images. Such assumptions can be that similar regions within object boundaries should have uniform disparity distribution. The global approach incorporates smoothness in the disparity estimation, which results in fewer errors caused by disparity discontinuities, occlusions and texture-less areas. The last step of disparity refinement acts as a post-processing step, where noise and uncertainties are removed and the disparity map is optimized to be more accurate. For example, occluded areas do not contain corresponding point matches since the occluded region is only visible within one of the images. Hence, occlusion filling can be used for estimating the disparities in these areas using adjacent values.

2.2.4 3D Reconstruction and Data Representation

Given the disparity map, the focal length and the baseline of the camera, 3D reconstruction can be performed to transform one of the image coordinate systems to the world coordinate system [12]. Each individual pixel coordinate is transformed from the image frame to the world frame using Equations (2.3), (2.4) and (2.5).

$$x = \frac{bX_2}{d} \tag{2.3}$$

$$y = \frac{bY_2}{d} \tag{2.4}$$

$$z = \frac{bf}{d} \tag{2.5}$$

Using Equations (2.3), (2.4) and (2.5), the 3D information from the images is recovered. The world coordinates are relative to the right camera's optical center but can similarly be expressed using the left camera's. The 3D information can be visualized using a point cloud. Since the points only describe physical distances and positions, there is no color information. Color can be applied using simple color mappings based on, for instance, the points' distances from the camera. Point clouds are generally computationally expensive to process. Thus, there is an option to use depth

images, which can be encoded with the same information as point clouds. Depth images can be used in conjunction with convolutional neural networks for extracting information.

2.3 Convolutional Neural Networks

A convolutional neural network (CNN) is a variant of the artificial neural network architecture [16]. CNNs are used for image-based machine learning. They are the current state-of-the-art analysis tool for examining and analyzing images. The network architecture can perform tasks such as object localization and classification. CNNs are predominantly made up of three components: convolutional-, pooling- and fully-connected layers. This structure is demonstrated in Figure 2.2, where each component occurs once. However, they are commonly stacked and repeated to build up a more complex architecture.

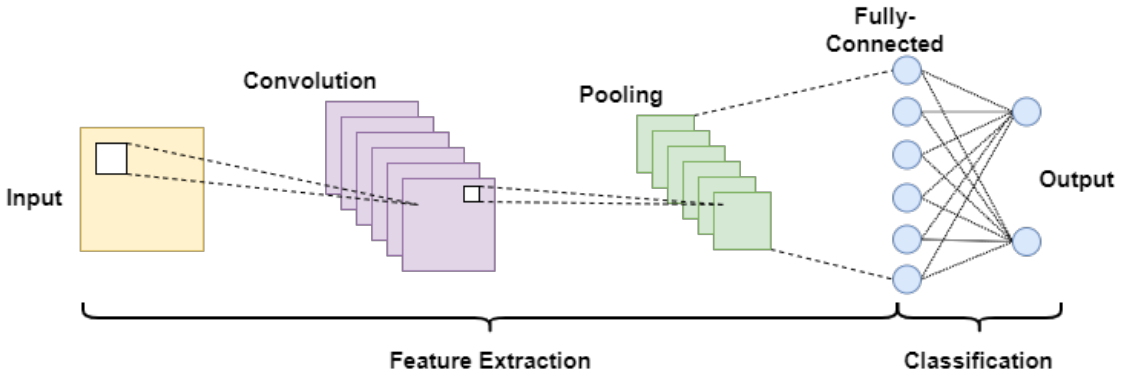


Figure 2.2: Simple CNN architecture with the three main components: convolutional-, pooling- and fully-connected layers. The convolutional layers slides a filter across the input, the pooling layer performs downsampling. Lastly, the fully-connected layer is used for evaluating the produced feature map for generating predictions.

The inputs to the network are images with three dimensions: spatial (width and height) and depth (channels). Input images commonly have three channels, corresponding to the RGB color format. Dimensions of the data are altered throughout the network by the component layers [17]. The first component, convolutional layers, performs convolution on the input with filters [16]. Furthermore, the filters have trainable parameters which are learned through backpropagation as training progresses. The general use of the convolutional layer is to extract features from the images, such as edges, patterns and more complex shapes. Pooling layers perform downsampling along the spatial dimension to reduce the complexity for succeeding layers. The outputs from convolutional- and pooling layers are commonly referred to as feature maps. Stacking multiple convolutional- and pooling layers results in a more complex feature map. Fully-connected layers are commonly used at the end of the network to evaluate the feature map.

A common use for CNNs is object detection in images [18]. Object detection consists of two steps: localization and classification. Localization embodies finding objects in the image, while classification identifies what the object is. Performing object detection in real-time has historically been a difficult task to solve. There now exist networks that manage object detection in real-time.

Training a neural network requires a great amount of data. A common challenge when using machine learning is gathering enough (labeled) data [19]. Transfer learning is a powerful method for neural networks which leverages knowledge of already trained networks to overcome the challenge. The concept involves adapting a network that has already been trained for a task on a new application. The network can then specialize on the new task with a smaller amount of data, compared to training it from scratch [20]. Network structures such as VGG-16 and ResNet-101 are commonly used within transfer learning [21].

2.4 The R-CNN Framework

A widely used object detection architecture is the region based convolutional neural network (R-CNN) family of convolutional networks, presented in [22], [23] and [24]. They perform both localization and classification, and as further developments have taken place, can predict in real-time. The variations of the R-CNNs are all based on the same concept. Firstly, the network generates region proposals, or guesses, of where objects are located within images. Secondly, the proposals are evaluated and classified. The main difference between the architectures within the R-CNN family is how region proposals are managed. The three versions: R-CNN, Fast R-CNN and Faster R-CNN, are summarized in Section 2.4.1.

2.4.1 Versions of R-CNN

The origin and the first version within the R-CNN family of networks is presented in [22]. It uses the selective search algorithm to generate thousands of region proposals per image. The proposals are fed through a CNN to extract features. Various networks can be used as feature extractor, such as VGG-16 [21]. The feature map is forwarded to a support-vector machine (SVM), which predicts if there is an object present in each proposed region. Furthermore, for each proposed region, a refinement of the bonding box is predicted to improve the precision of the box. There are two main issues with the method: computational speed and training. R-CNN is not practical to run in real-time, as passing thousands of individual proposals through the feature extractor takes a long time as no computations are shared. Furthermore, the network is not trained end-to-end, since the selective search algorithm is fixed.

The computational speed is improved in the second version of the R-CNN framework, named Fast R-CNN [23]. The main change from the previous iteration is how the region proposals are handled. Rather than feeding the network with thousands of proposals separately, the entire image is fed directly to the CNN together with the set of region proposals from the selective search algorithm. The region proposals

are extracted from the feature map, rather than the input image. Region of interest (ROI) pooling is performed on the produced feature map to extract regions of fixed shape. The result is fed into fully-connected layers for evaluation. The convolution operation is performed only once, compared to processing thousands of proposals individually. This reduces the number of operations and improves the computational speed [23]. Even though this decreases the computational time per image significantly, selective search is still used and remains the bottleneck for computational performance.

The third and latest version of the algorithm further improves the efficiency and is called the Faster R-CNN architecture [24]. The authors solved the bottleneck caused by the selective search in Fast R-CNN by developing and using a region proposal network (RPN) instead. By utilizing the RPN, the algorithm can run in real-time, contrary to its predecessors. Similar to Fast R-CNN, the output of the proposal method is passed along to ROI-pooling and a classifier.

All three versions are depicted in Figure 2.3. RPN and ROI-pooling are explained in Sections 2.4.2 and 2.4.3 respectively.

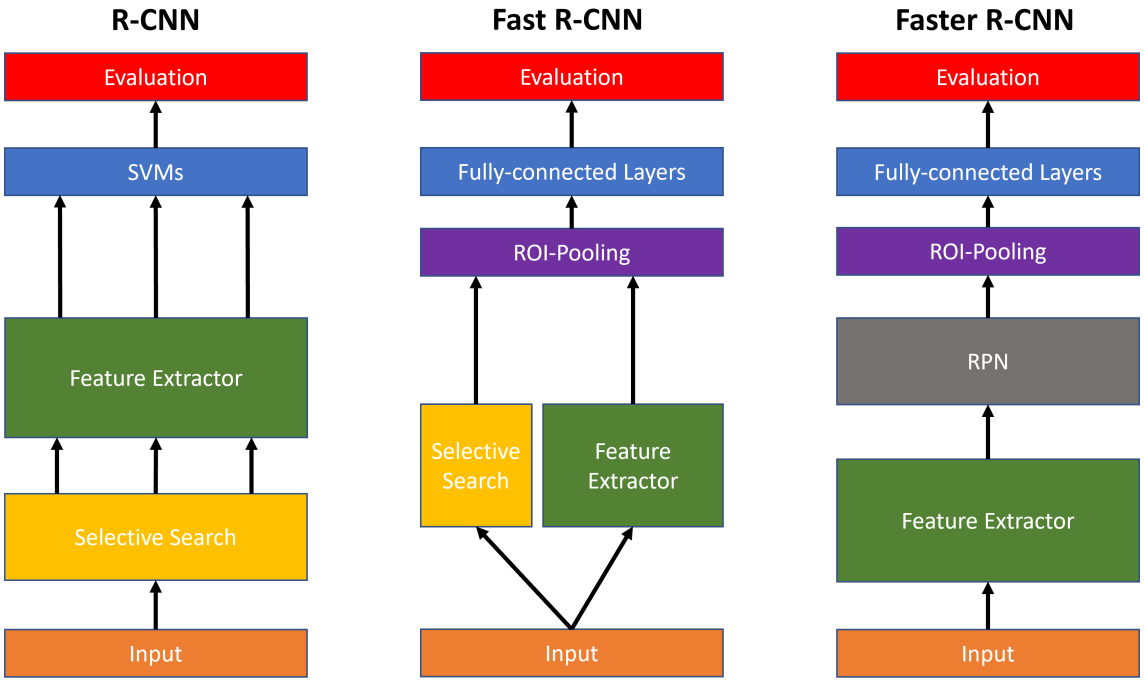


Figure 2.3: Overview of structure and components of the three versions within the R-CNN family. The main difference is the handling of region proposals. The R-CNN and the Fast R-CNN use the fixed selective search to generate proposals, compared to the Faster R-CNN which uses a region proposal network (RPN). Furthermore, depending on the version, a special region of interest (ROI) pooling layer is required before evaluation. Finally, evaluation is done through support-vector machines (SVMs) or fully-connected layers.

2.4.2 Region Proposal Network

The region proposal network (RPN) is introduced by Ren et. al. in [24]. As the name suggests, the RPN generates proposals where objects may potentially be located in images. It uses a fixed amount of bounding boxes, which are the same for every image, as a basis for its guesses. The RPN predicts the offset required for each bounding box to fit an object, as well as an objectness score, indicating if the box contains an object (*foreground*) or not (*background*).

The fixed bounding boxes are generated through a grid of points, called anchors. Multiple boxes are generated at each anchor with different scales and ratios. The procedure of how anchor boxes are generated based on scales and aspect ratios is illustrated in Figure 2.4.

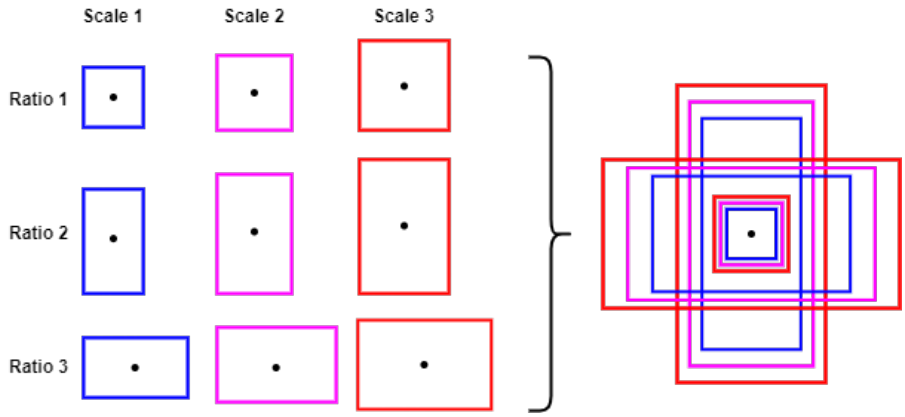


Figure 2.4: Box scales and aspect ratios are combined to generate anchor boxes of various shapes and sizes, as depicted to the right. These boxes are generated for each anchor (black dot) in the grid.

In the original paper, three scales and three ratios are used to produce a total of nine boxes for each anchor [24]. Anchor boxes are evaluated against the ground truth boxes using the intersection over union (IOU) metric. IOU is an important metric used in both training and evaluation. It is used to evaluate the similarity of two bounding boxes, and is computed according to the illustration in Figure 2.5.

$$\text{IoU} = \frac{\text{Area of Intersection of Two Boxes}}{\text{Area of Union of Two Boxes}} =$$

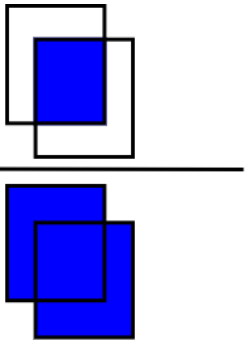


Figure 2.5: The intersection over union (IOU) is the intersecting area of the boxes divided by the union area of the boxes.

In the RPN, binary class labels are assigned to each box for training. The assigned labels are based on the IOU with the ground truth box. Table 2.1 present the IOU boundaries for foreground and background boxes. Anchor boxes that are considered neither foreground or background are ignored and do not contribute to training the network [24].

Table 2.1: Corresponding labels for IOU values.

	$IOU > 0.7$	$0.3 \leq IOU \leq 0.7$	$IOU < 0.3$
Label	Foreground	Ignored	Background

In the cases where no sample fulfills the condition for the foreground label in Table 2.1, the sample with the highest IOU is chosen.

The training is performed in mini-batches, where one mini-batch arises from a single image. Each mini-batch contains many foreground and background anchor boxes that can all be used for training. However, each image is generally dominated by background samples, and the training would be biased toward these. Hence, a subset of anchors is chosen randomly to be used for the training. In the original paper, 256 anchors are chosen with an equal ratio of foreground and background anchors. If there are less than 128 foreground samples, the mini-batch is padded with more background samples.

2.4.3 ROI-Pooling

The initial set of proposed boxes is shifted with the offset, predicted by the RPN, to obtain new refined boxes. These boxes have various sizes, but are required to have a fixed size for the final classification and regression layers. Therefore, a special pooling layer is used called region of interest (ROI) pooling. The ROI pooling layer transforms every region of interest defined by the proposal boxes to the same size, independent of input size, visualized in Figure 2.6.

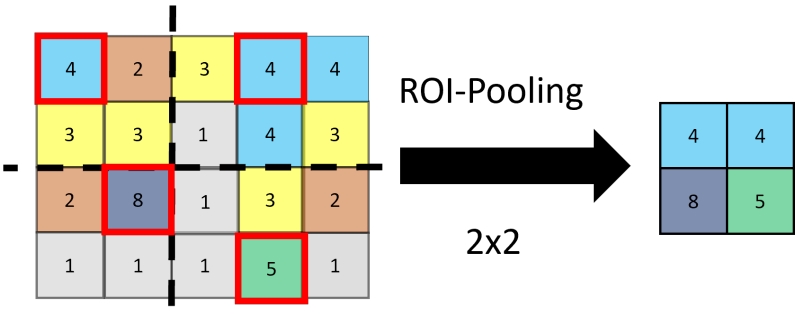


Figure 2.6: 2x2 region of interest (ROI) pooling. The input is divided in a 2x2 grid and the maximum of each area, marked with red squares, is the output value.

All regions of interest are divided into a fixed-sized grid. Max-pooling is performed within each area within the grid. It is an operation that returns the maximum value

within a specified area. The output of ROI pooling will always be of fixed size. It can then be fed into fully-connected layers for evaluation for object classification and bounding box regression.

2.4.4 Evaluation and Loss Functions

Training performance for the entire Faster R-CNN framework is evaluated through four loss functions. The RPN contains two: one for the regression layer that performs bounding box regression, and one for the classifier that determines if a bounding box is foreground or background. The final layers of the Faster R-CNN have the remaining two: one for the regression layer that further refine the bounding boxes from the RPN, and one for the classifier that classifies the objects in the foreground boxes. Both pairs of loss functions use the same metrics: a cross-entropy loss for the classification layers and the smooth L_1 loss for the regression layers. The loss functions are presented in Equations (2.6) and (2.7), in accordance with [24].

$$L_{CE}(p_i^*, p_i) = -\sum_{j=0}^{N_{class}-1} p_i^* \log p_i \tag{2.6}$$

$$L_1(t_i^*, t_i) = \begin{cases} \frac{1}{2}(t_i^* - t_i)^2, & \text{if } |t_i^* - t_i| \leq 1 \\ (|t_i^* - t_i| - 0.5), & \text{otherwise} \end{cases} \tag{2.7}$$

In Equation (2.6) and (2.7), i is the index of a box in the current mini-batch. Furthermore, the set of p_i is the class probabilities for each box. The predicted shift is contained in t_i. It describes the offset for the center, width and height of the box to align with a ground truth box. The number of classes is denoted N_{class}. Ground truth values are denoted with an asterisk.

For the binary case, the crossentropy in Equation (2.6) can be written as Equation (2.8).

$$L_{BCE}(p_i^*, p_i) = -(p_i^* \log p_i + (1 - p_i^*) \log(1 - p_i)) \tag{2.8}$$

For the RPN, the binary cross-entropy function is used and the labels are foreground (1) and background (0). Ground truth for the regression is the offset for each anchor box to a true box. For the final layers, the categorical cross-entropy function is used and the labels are multiple classes. Offsets to the same true boxes are used for the regression. However, the offset values are not the same, since the anchor boxes have been shifted by the RPN prediction. The loss function for a mini-batch is defined as the sum of losses from each anchor, according to Equation (2.9).

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{CE}} \sum_i L_{CE}(p_i, p_i^*) + \frac{1}{N_{reg}} \sum_i p_i^* L_1(t_i, t_i^*) \tag{2.9}$$

The regression loss is only activated for foreground anchors. Finally, the loss terms are normalized using N_{CE} and N_{reg} corresponding to the mini-batch size and the number of anchor locations.

2.4.5 Summary of Faster R-CNN

To summarize, the full Faster R-CNN structure is illustrated in Figure 2.7.

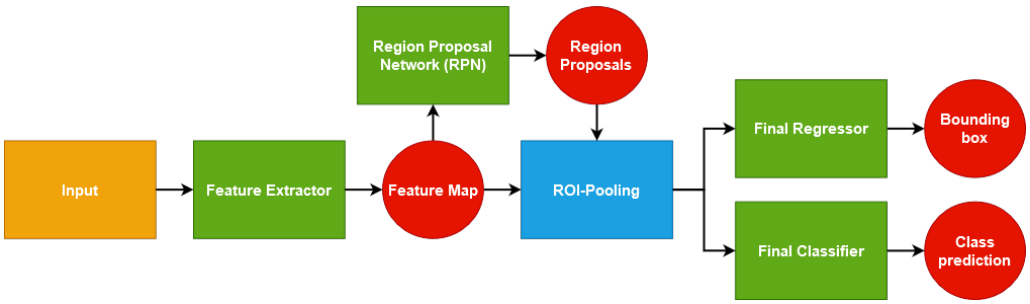


Figure 2.7: All components and connections to build the Faster R-CNN structure.

The input is fed to a feature extractor, often a pre-trained network. The resulting feature map is forwarded to an RPN to identify regions that may contain an object in the image. ROI-pooling is performed on the feature map together with the region proposals. Finally, the proposals are evaluated.

3

Methods

This chapter presents the methods and workflow to investigate optical load detection. The method includes hardware choice and setup, data acquisition, data processing with neural networks and evaluation.

3.1 Hardware

A vital part of the load detection solution was the choice and mounting of hardware. Data acquisition, data processing and evaluation depend directly on the choices regarding hardware. This section contains motivations of chosen hardware, concerning characteristics and desired data. Furthermore, the hardware integration including settings and development of the data acquisition is presented.

3.1.1 Choice of Equipment

The idea was to use depth data for fill-factor estimation and RGB data for the identification of the material type. Depth data was of interest since it contains 3D information appropriate for determining bucket fill-factor, which might not be available in regular 2D RGB images. Depth information can be obtained through various non-contact techniques involving 3D imaging and range measurements, some of them mentioned in the theory in Section 2.1.

The choice of sensor for the developed system was a stereo camera. One large benefit is that it captures both RGB and depth data simultaneously. A stereo-vision system captures images from two perspectives to retrieve 3D information using triangulation. Capturing images using a camera is simple and fast. However, the accuracy and the processing time of the depth calculation depend on the used algorithms and the image content. For example, high accuracy can be difficult to achieve when capturing surfaces with ambiguous textures.

The sensor used was an Intel RealSense Depth Camera D435i with integrated Intel RealSense Depth Module D430, a full-HD (1920×1080) RGB camera, an infrared projector and an inertial measurement unit (IMU) [25]. The depth module uses stereo vision for calculating depth. An infrared projector projects a static infrared pattern to improve depth accuracy in scenes with ambiguous textures. The baseline

of the D435i sensor is 50 mm, focal length 1.93 mm and optimal operating range is specified to the interval 0.3 m to 3 m [26].

3.1.2 Camera Settings and Data Acquisition Pipeline

Intel provides a software developer kit for convenient operation of their depth cameras [27]. Furthermore, an application programming interface (API) is provided which was used for streaming and acquisition of data. Initializing camera settings, starting and displaying the camera stream and capturing data can all be done through the API. The acquisition pipeline stored an RGB image and the depth data. The RGB image and the depth data had to be aligned since they were captured from slightly different perspectives due to the placement of the sensors. RGB images and depth data were captured with a 1920×1080 and a 1280×720 resolution respectively, at 30 frames per second (FPS). In the alignment step, the depth data was upsampled to match the resolution of the RGB image.

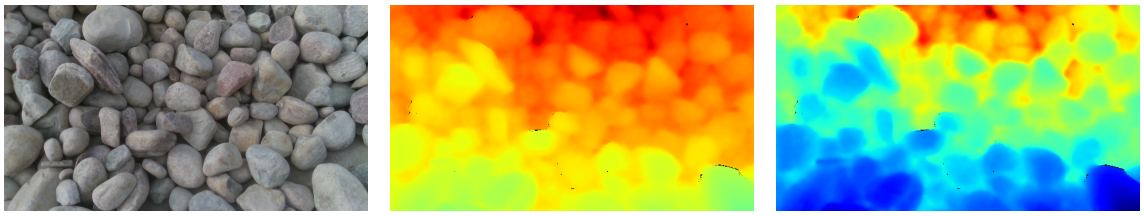
The Intel RealSense Depth Camera D435i has an integrated processor with built-in algorithms for processing the data. The processor provides fast depth calculations but the specific algorithms used are not publicly available. However, there are plenty of settings available for adapting the performance for the specific application. Some setting presets are available, where *High Density* was used [28]. It is the recommended preset for object recognition and enhanced 3D photography.

3.2 Depth Image Generation

Depth data was used for generating depth images. Distance values outside of a specified interval were filtered, since these correspond to measurements outside the range of the container. A color mapping was applied, called *Jet* [29]. The color mapping is depicted in Figure 3.1. It shows that points close to the sensor tend towards blue while points further away tend towards red. Filtered points are mapped to black.

Figure 3.1: The color map *Jet*, used for colorizing depth data [29]. Low distance values are mapped to blue, while higher values tend towards red.

The distance interval had to be the same for all data captured with a sensor and bucket setup. This was to ensure the data was processed consistently and was comparable. All distance values within the interval were normalized and the color map was applied. A tighter distance interval provides larger contrast in terms of depth differences in the images, emphasizing the structure of the measured material. Figure 3.2 depicts examples of generated depth images using two distance intervals for comparison.



(a) RGB image **(b)** Large interval **(c)** Tight interval

Figure 3.2: Examples of generated depth images and comparison of filtering interval. The image to the left is the the regular RGB image displaying the scene, middle image is a produced depth image with a large distance interval and the right image is a generated depth image with an adapted and tighter distance interval.

The examples in Figure 3.2 depicts cobblestone with large variations in terms of structure. Comparing the depth images in Figures 3.2b and 3.2c, the tighter interval provides higher contrast and more details.

3.3 System Architecture

This section describes the system architecture used for estimating fill-factor, volume, type, weight and where in the images the loaded material is located. An overview of the system is depicted in Figure 3.3.

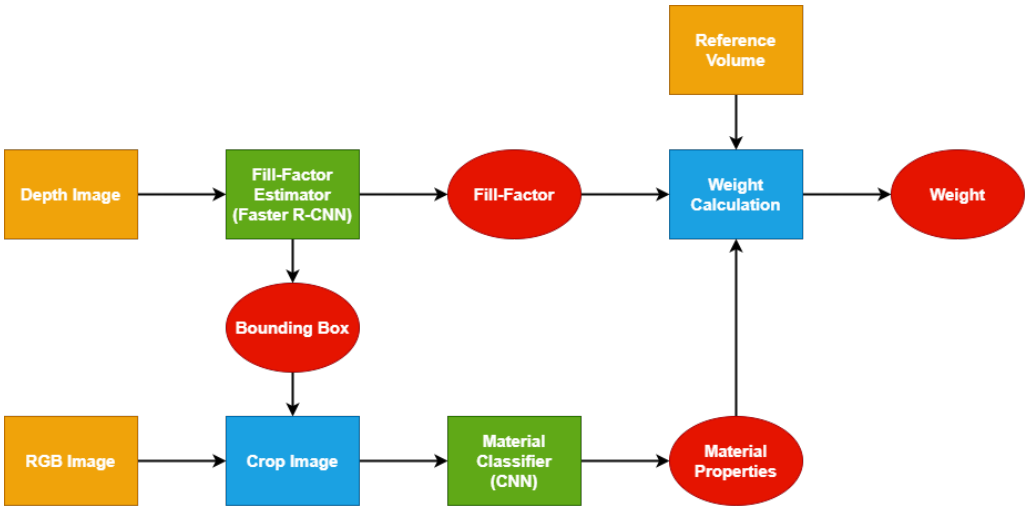


Figure 3.3: An overview of the load detection system describing the flow and processing of the data. By utilizing a depth and RGB image, the weight of loaded material can be calculated. The system uses CNN to determine both the fill-factor and material type. Combining this information with a predetermined reference volume and density results in a weight prediction.

The system in Figure 3.3 received two images as input: a depth image and an RGB image. Depth images were used for material localization and fill-factor estimation.

Passing the depth image through a Faster R-CNN resulted in a bounding box enclosing the bucket and material, as well as a predicted fill-factor. The bounding box was used to mark the area in the RGB image where the material classifier predicts the material type. By knowing the material type, material properties can be retrieved. The weight was calculated by combining the density of the material, the predicted fill-factor and the reference volume based on the bucket capacity. This is further explained in Figure 3.4.

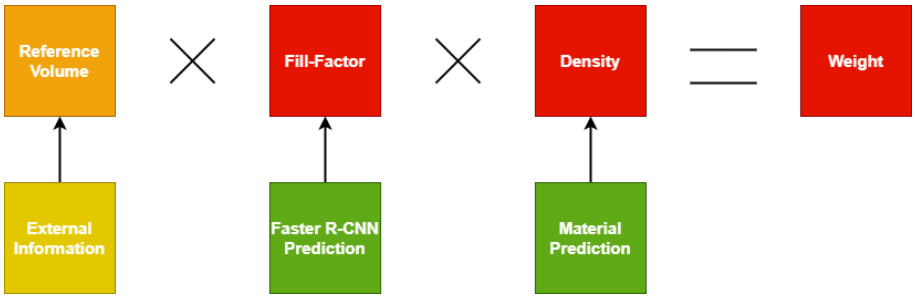


Figure 3.4: Weight estimation using fill-factor, reference volume, and material prediction. The reference volume is external information specified on the bucket, the fill-factor is estimated through the Faster R-CNN and the material density is extracted from a database based on the prediction from the material classifier.

Sections 3.3.1 and 3.3.2 describes in detail the steps in the system architecture, in specific the fill-factor estimation and the material classification.

3.3.1 Fill-Factor Estimation

A Faster R-CNN architecture was used to localize the material and classify the fill-factor of the container. A pre-trained VGG-16 network was used as feature extractor. The output of the network was a single bounding box and probabilities corresponding to intervals of fill-factors. The fill-factor represents to what degree the container is filled. It can be calculated through both weight and volume, and are equivalent given a similar density, regardless of volume. For training the network, a weight to fill-factor association had to be established for the sensor and bucket setup, due to reference data being weights. The density of the material, ρ, was used together with the bucket volume capacity, V_{ref}, for calculating a reference weight w_{ref} in Equation (3.1).

$$w_{ref} = \rho V_{ref} \tag{3.1}$$

The reference weight obtained through Equation (3.1) was used to relate measured weights to fill-factors. Since the desired output x_{est} was a continuous value, a weighted sum was calculated using the probabilities p_i and the mid-points z_i of the intervals across all classes N_{class}, presented in Equation (3.2).

$$x_{est} = \sum_{i=1}^{N_{class}} p_i z_i \tag{3.2}$$

3.3.2 Material Classification

The purpose of the material classifier was to detect the type of the handled material and retrieve the density. Similar to the fill-factor estimation, the material classifier utilized transfer learning with a pre-trained VGG-16 network as a feature extractor. The RGB image captured by the stereo camera was cropped using the predicted bounding box from the fill-factor network. The cropped image was then passed through the feature extractor. Lastly, the resulting feature map was fed through a set of fully-connected layers and a classification layer to predict the material type.

The network predicts probabilities over the material categories. When training the network, the categories of interests had to be decided. For certain applications it may be sufficient to detect the class of the material, such as gravel. In others, it can be necessary to identify further details, such as which fraction of the gravel it is. Therefore, predictions were evaluated in both groups of materials and as individual classes.

3.4 Data Acquisition

Convolutional neural networks were utilized in the developed system. Hence, a large quantity of labeled data was required for training. The following sections describe how various data acquisition campaigns were conducted.

3.4.1 Initial Investigation

An initial investigation was conducted to evaluate the feasibility of the proposed setup, given ideal conditions. To obtain maximal depth information and avoid occlusions, the camera was mounted right above a container holding some material. Figure 3.5 shows the setup together with an example RGB and depth image generated from depth data, captured by the stereo camera.

(a) Setup overview **(b)** RGB image **(c)** Depth image

Figure 3.5: Initial static camera setup for feasibility evaluation. Left image depicts a side view of the setup, the middle image the field of view captured from the RGB camera and the image to the right the produced depth image.

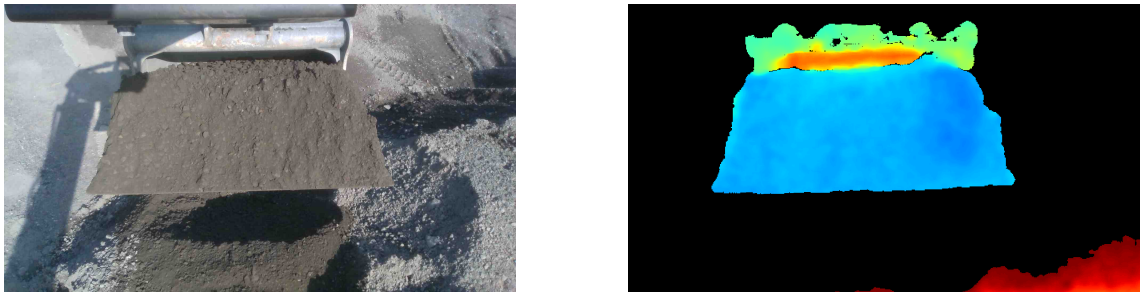
Various quantities of gravel were used for all measurements. To ensure the only variations in the measurements were the fill-factor of the container, the camera and container positions were not altered. For each measurement, the material and container were weighed using a scale. These weights were used as ground truth references. The largest weight within the dataset was used for converting the ground truth weights to fill-factors.

3.4.2 Excavator

The second acquisition campaign was conducted on an excavator. A mounting platform with magnets was constructed for convenient mounting of the camera on the machine. The platform was mounted on the stick, as shown in Figure 3.6.

Figure 3.6: Intel RealSense Depth Camera D435i mounted on the excavator, used for collecting data.

A similar type of gravel was used as in the initial investigation. During the data acquisition, the pose of the bucket was approximately the same for each measurement, which limited the number of influential factors in the resulting data. Figure 3.7 depicts examples of RGB and produced depth images using the sensor setup.



(a) RGB image **(b)** Depth image

Figure 3.7: Sample images captured by the stereo camera mounted on the excavator. The left is the RGB image and the right the generated depth image.

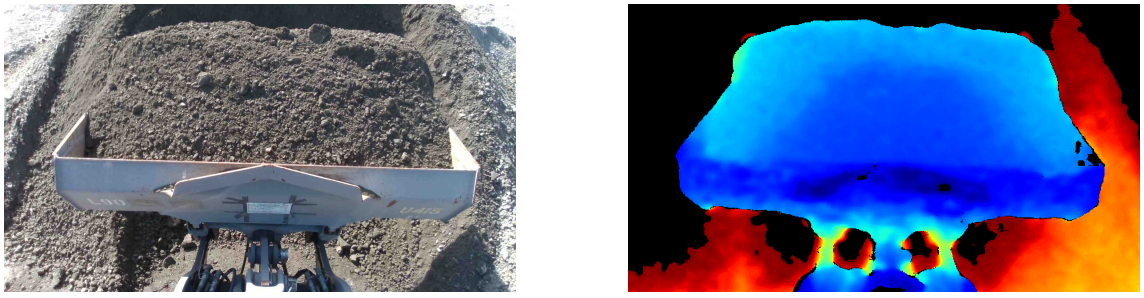
As ground truth reference, an on-board weighing system in the excavator was used. The density of the gravel was retrieved by weighing a known volume of it. A reference weight was then obtained through Equation (3.1). The maximum weight within the dataset, the number of measurements and the calculated reference weight are found in Table 3.2.

3.4.3 Wheel Loader

The third campaign was conducted on a wheel loader. An extension was constructed and attached to the cabin roof for mounting the stereo camera. The extension reached forward and upward to grant the sensor a field-of-view of the bucket, as well as protect the sensor from the boom and the bellcrank of the wheel loader. The setup is depicted in Figure 3.8.

Figure 3.8: Intel RealSense Depth Camera D435i mounted on the wheel loader, used for collecting data.

The same pile of gravel was used as in the excavator campaign. Figure 3.9 depicts one of the conducted measurements.



(a) RGB image **(b)** Depth image

Figure 3.9: Sample images captured by the stereo camera mounted on the wheel loader. The left is the RGB image and the right the generated depth image.

Similar to the excavator, an on-board weighing system was used to retrieve ground truth references. A reference weight was calculated using Equation (3.1). The maximum weight within the dataset, the number of measurements and the calculated reference weight are found in Table 3.2.

3.4.4 Material Collection

For material classification, a large quantity of regular RGB images of various material types was required. Materials were photographed at a material dealer, offering stacked piles of various kinds such as macadam, gravel and sand. 50 images were captured from assorted perspectives for each material type, using the Intel RealSense Depth Camera. Furthermore, materials were weighed to build a dataset of densities. Figure 3.10 depicts some piles used during the material data acquisition.

Figure 3.10: Examples of piles used for acquiring RGB images used for training the material classification network.

3.5 Training

The Faster R-CNN was trained on depth images, which were produced through the steps described in Section 3.2. The intervals used for the established datasets were fine-tuned manually and presented in Table 3.1.

Table 3.1: Distance intervals used for filtering depth data.

	Static test	**Excavator**	**Wheel loader**
Min distance (m)	0.9	1.4	2.0
Max distance (m)	1.45	3.1	4.2

Within each image, the container with the material was annotated with a bounding box, used for training the Faster R-CNN. Data augmentation was applied to expand the training dataset by slightly altering the images with combinations of rotations and filters. The depth images were resized to reduce the computational cost for the network, without losing important details in the images. Henceforward, the data was divided into a training and a test set for evaluation. Table 3.2 summarize the sizes of the datasets and hyperparameters used for training the fill-factor estimator (Faster R-CNN).

Table 3.2: Sizes of datasets used and hyperparameters used for training Faster R-CNN.

	Static test	**Excavator**	**Wheel loader**
Training images	180	386	280
Training images (with augmentations)	1 620	3 474	2 133
Test images	20	69	43
Number of categories	10	12	15
Image size (resized)	$300, 533$	$300, 533$	$300, 533$
Anchor box scales	$150, 180, 200$	$190, 200, 210$	$190, 200, 210$
Anchor box ratios	1:1, 1:1.2, 1.2:1	1:1, $\frac{1}{\sqrt{2}}$:$\frac{2}{\sqrt{2}}$, $\frac{2}{\sqrt{2}}$:$\frac{1}{\sqrt{2}}$	1:1, $\frac{1}{\sqrt{2}}$:$\frac{2}{\sqrt{2}}$, $\frac{2}{\sqrt{2}}$:$\frac{1}{\sqrt{2}}$
Ref. weight [kg]	65	1 394	4 420
Max weight [kg]	65	1 670	6 520

Note, since the reference weight was less than the maximum weight for the excavator and the wheel loader campaigns, fill-factors over 1.0 occurred. Furthermore, predicted weights below 200 kg for the excavator and the wheel loader were considered to be empty buckets since no measurements were conducted below this weight. For the material classification, images are divided into datasets according to Table 3.3.

Table 3.3: Dataset sizes and number of classes used in the material classification network.

Parameters	Quantity
Training images	595
Training images (with augmentations)	4 760
Test images	171
Number of materials	17

3.6 Evaluation

Various metrics were used for evaluating the fill-factor predictions. Approximation or sample errors is a rudimentary metric based on the differences between the ground truth and the predicted values. The mean of the approximation errors are calculated as mean absolute error (MAE) and mean absolute percentage error (MAPE), according to Equations (3.3) and (3.4) respectively.

$$\text{MAE} = \frac{1}{N}\sum_{i=0}^{N-1} \mid x_i^* - x_i \mid \tag{3.3}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=0}^{N-1} \mid \frac{x_i^* - x_i}{x_i^*} \mid \tag{3.4}$$

In Equations (3.3) and (3.4), x^* and x are the ground truth and predicted values respectively, and N is the total amount of samples. An option is to calculate the MAPE relative the maximum value within the dataset, x_{max}^*, to not punish large relative errors caused by predictions of low values. The metric is denoted MAPE_{max} and is presented in Equation (3.5).

$$\text{MAPE}_{max} = \frac{1}{N}\sum_{i=0}^{N-1} \mid \frac{x_i^* - x_i}{x_{max}^*} \mid \tag{3.5}$$

The relative error can only be obtained for non-zero ground truth values, which is not possible when including empty buckets in the evaluation. Additional metrics was used to incorporate all data such as standard deviation (STD or σ), which is calculated using the mean (μ) according to Equations (3.6) and (3.7).

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i^* - x_i \tag{3.6}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(x_i - \mu)^2} \tag{3.7}$$

The distribution of the data was described with the mean and the STD. For instance, 95% of the data is within two standard deviations (2σ) from the mean, assuming the data is normally distributed. It was used for representing within which interval most of the errors were contained.

The overall performance of the system was directly influenced by how well the regions of interest (bounding boxes) are found. Therefore, the mean IOU was calculated for all bounding boxes as an indication of how well the predicted bounding boxes match the ground truth. The calculation of IOU is illustrated in Figure 2.5. The bounding boxes were visually inspected to further evaluate if the predictions of bucket and material location are valid.

The material classification was evaluated using the accuracy of class predictions. The accuracies for all predictions can be summarized in a confusion matrix. It evaluates true and predicted classes in a grid-like fashion. Each entry is calculated as the relative frequency across each true class. Ideally, the diagonal should be ones, while off-diagonal values should be zero. In such a case, the predicted class coincides perfectly with the true class. Values occurring on the off-diagonal are the incorrect predictions, which provide insight into which classes the network confuses. A schematic of the confusion matrix with some example predictions is found in Figure 3.11. In the example, material 2 was misclassified with material 1 and material 3 20% of the time respectively, while material 3 was misclassified as material 2 10% of the time.

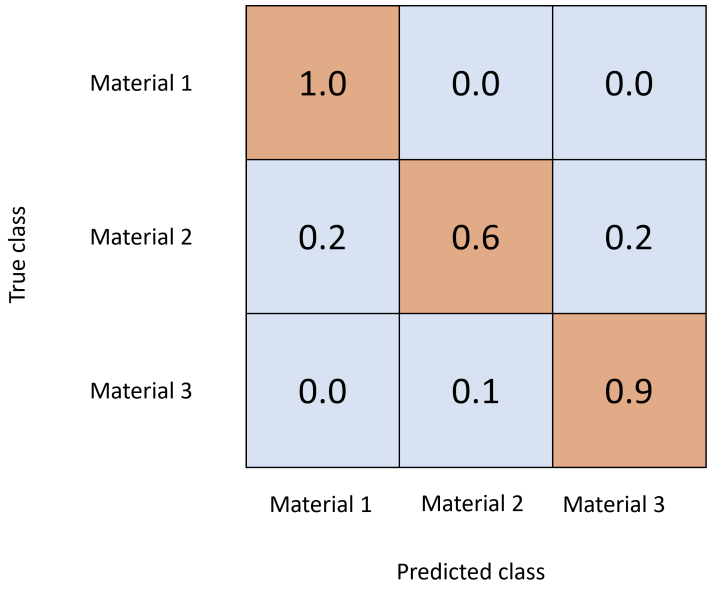


Figure 3.11: An example confusion matrix. Material 1 was predicted correctly, while material 2 and 3 had some misclassifications.

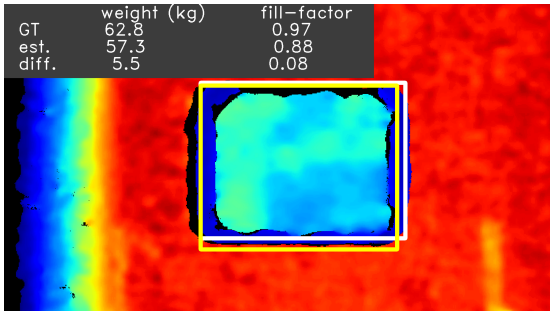
4

Results

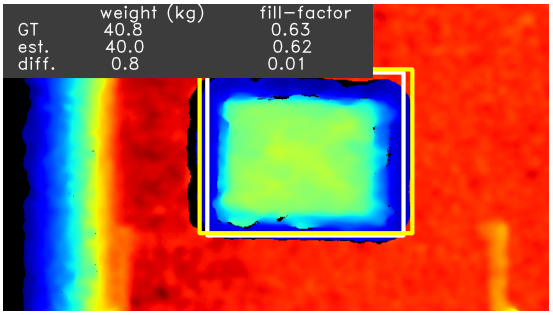
This chapter presents results from fill-factor and weight estimations from the trained network, using the data collected during the acquisition campaigns. Furthermore, the results from material classifications are presented.

4.1 Fill-Factor and Weight Estimations

The results for the fill-factor and weight estimations are presented individually for each conducted acquisition campaign. Details about the datasets and parameters used for training the network are available in Table 3.2. Figures 4.1-4.3 presents two images from each campaign with their corresponding bounding box and fill-factor predictions.



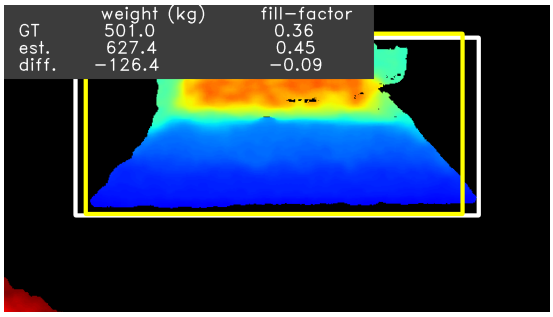
(a) One of the largest deviations.



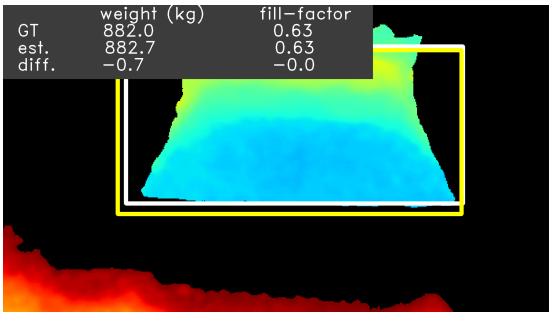
(b) One of the best predictions.

Figure 4.1: Two samples from the *static campaign* with ground truth (GT) and predictions (est.). Ground truth bounding boxes are drawn in white and predicted bounding boxes in yellow. The estimated fill-factors and the resulting weight estimations are presented together with the sample error (diff.).

From Figure 4.1, it is observed that the depth images are similar in appearance. The differences are the depth variations within the containers, where higher fill-factor results in a more blue nuance.



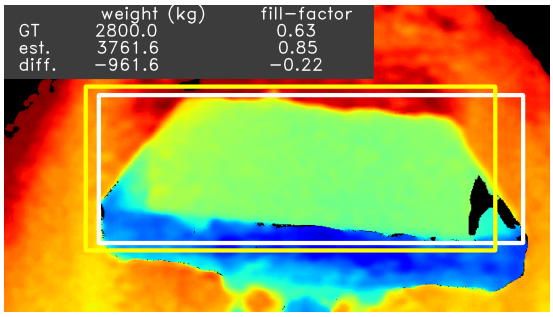
(a) One of the largest deviations.



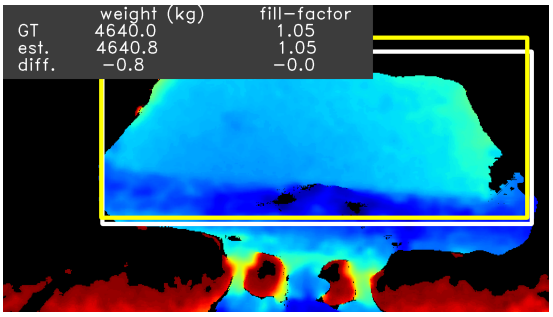
(b) One of the best predictions.

Figure 4.2: Two samples from the *excavator campaign* with ground truth (GT) and predictions (est.). Ground truth bounding boxes are drawn in white and predicted bounding boxes in yellow. The estimated fill-factors and the resulting weight estimations are presented together with the sample error (diff.).

In Figure 4.2, the bucket is distinguished from the background through the depth filtering. In Figure 4.2b, some of the background is included due to the bucket being close to the material pile or ground. The orientations of the buckets are noticeable through their profiles in the images. A curled bucket results in a wider and non-uniform profile while a bucket parallel to the sensor is more rectangular.



(a) One of the largest deviations.



(b) One of the best predictions.

Figure 4.3: Two samples from the *wheel loader campaign* with ground truth (GT) and predictions (est.). Ground truth bounding boxes are drawn in white and predicted bounding boxes in yellow. The estimated fill-factors and the resulting weight estimations are presented together with the sample error (diff.).

In Figure 4.3, outlier rejection is noticed to be more difficult since much of the background is included in the depth images. Furthermore, the bucket is not centered in the sample images. Compared to the other setups, the bucket moves laterally. However, the material is still localized properly.

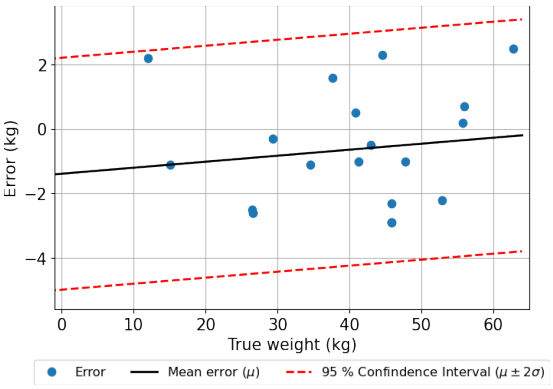
It is observed for each sample in Figures 4.1-4.3 that the predicted bounding boxes visually overlap to a large extent with the true reference boxes. The predicted fill-factors are used to calculate the estimated weights according to the methodology

established in Section 3.3. Evaluating the test images using the metrics described in Section 3.6 yields the results presented in Table 4.1.

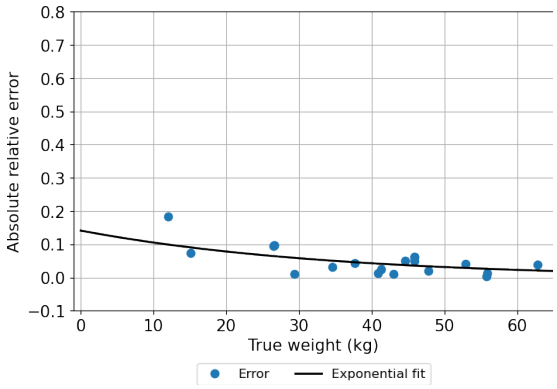
Table 4.1: The performance of the fill-factor and weight estimations, evaluated through presented metrics.

Metric \ Campaign		Static	Excavator	Wheel loader
Fill-factor	MAE	0.03	0.04	0.05
	STD	0.03	0.05	0.06
Weight	MAE [kg]	1.8	55.3	199.2
	STD [kg]	2.1	70.6	258.8
	MAPE [%]	5.1	8.9	6.0
	MAPE_{max} [%]	2.5	3.3	3.0
	Mean IOU	0.86	0.82	0.83

In Table 4.1, the MAE and the STD are presented individually for fill-factor and weight, while relative metrics and mean IOU are independent of data type. It is observed that the static setup yields the least deviating predictions considering all metrics. The MAE and STD in terms of fill-factor are similar for the excavator and the wheel loader. However, since the wheel loader accumulate more material due to a larger volume capacity of the bucket, the errors in absolute weights are larger. Nonetheless, both MAPE and MAPE_{max}, are less for the wheel loader compared to the excavator. The mean IOU further confirms the observation that the bounding boxes overlap to a large extent with the reference boxes. Supplementary results in terms of sample and relative errors, with regard to the loaded weight, are presented in Figures 4.4, 4.5 and 4.6.

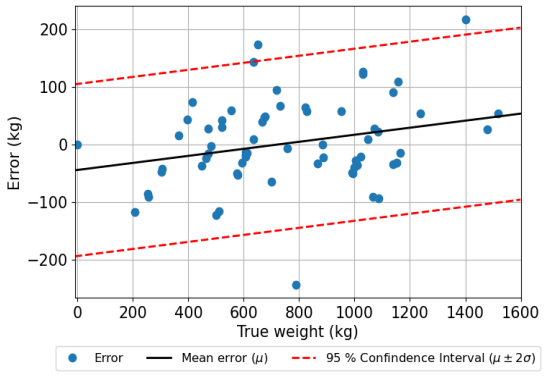


(a) Sample error $x^* - x$ with mean and confidence interval.

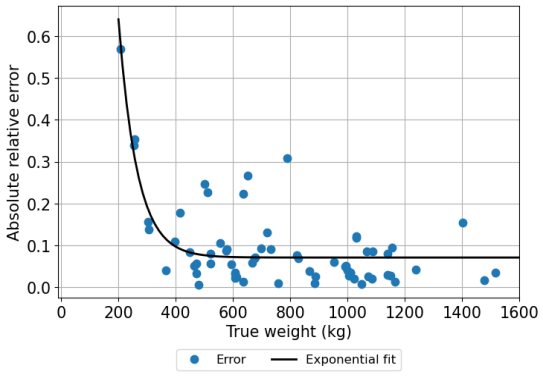


(b) Absolute relative error fit to a decaying exponential.

Figure 4.4: Sample and absolute relative errors with respect to loaded (true) weights for the *static campaign*.

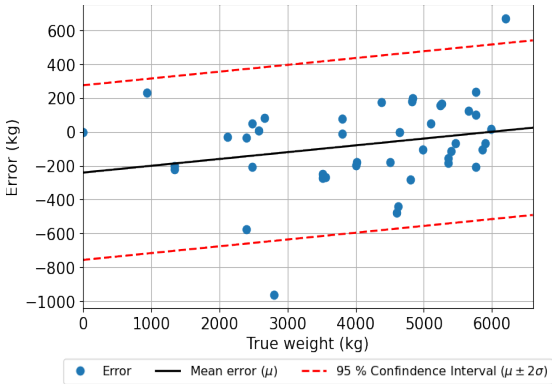


(a) Sample error $x^* - x$ with mean and confidence interval.

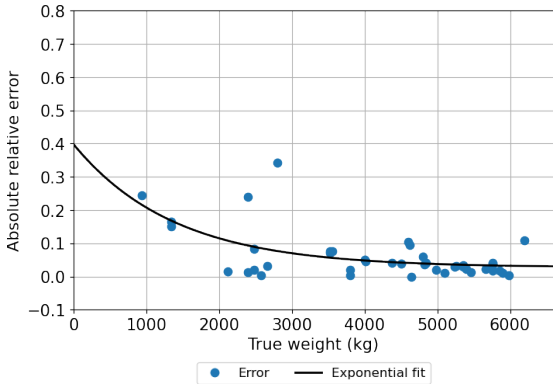


(b) Absolute relative error fit to a decaying exponential.

Figure 4.5: Sample and absolute relative errors with respect to loaded (true) weights for the *excavator campaign*.



(a) Sample error $x^* - x$ with mean and confidence interval.



(b) Absolute relative error fit to a decaying exponential.

Figure 4.6: Sample and absolute relative errors with respect to loaded (true) weights for the *wheel loader campaign*.

In Figures 4.4a, 4.5a and 4.6a, the sample errors, $x^* - x$, are depicted together with the mean and the two standard deviation confidence intervals. A negative sample error is an overestimate of the weight whereas a positive error is an underestimation. The slope of the drawn mean error indicates how the sample errors correlate with loaded weights. It is noticed that for low weights, the system tends to overestimate, and for higher weights, underestimate. Moreover, in Figures 4.4b, 4.5b and 4.6b, a decaying exponential is fit to the relative errors. The exponential shows the trend of the relative errors, which is observed to decrease as the loaded weight increases.

The sum of the sample errors is also interesting. It indicates how the system performs over multiple measurements. Table 4.2 presents the sum of the sample errors, a sum of the loaded weights and the relative error, for each campaign. The number of samples for each campaign is presented in Table 3.2.

Table 4.2: Sum of sample errors for each campaign together with the sum of the loaded weights and relative error.

	Static test	Excavator	Wheel loader
Sum of errors [kg]	−13	−260	−3 500
Total loaded weight [kg]	790	52 700	176 000
Relative error [%]	−1.6	−0.5	−2.0

From Table 4.2, it is observed that throughout all campaigns, the system overestimates the weight. For the most part, errors tend to even out, resulting in a low relative error.

4.2 Material Classification

The second part of the load detection system is the material classification. The full list of materials used for training and evaluating the developed system is available in Appendix A, together with material sample images and densities. For convenience and visualization purposes, the materials are categorized into seven main categories based on material properties. The predictions are conducted for the individual material types, where the categorized predictions are presented in the confusion matrix in Figure 4.7.

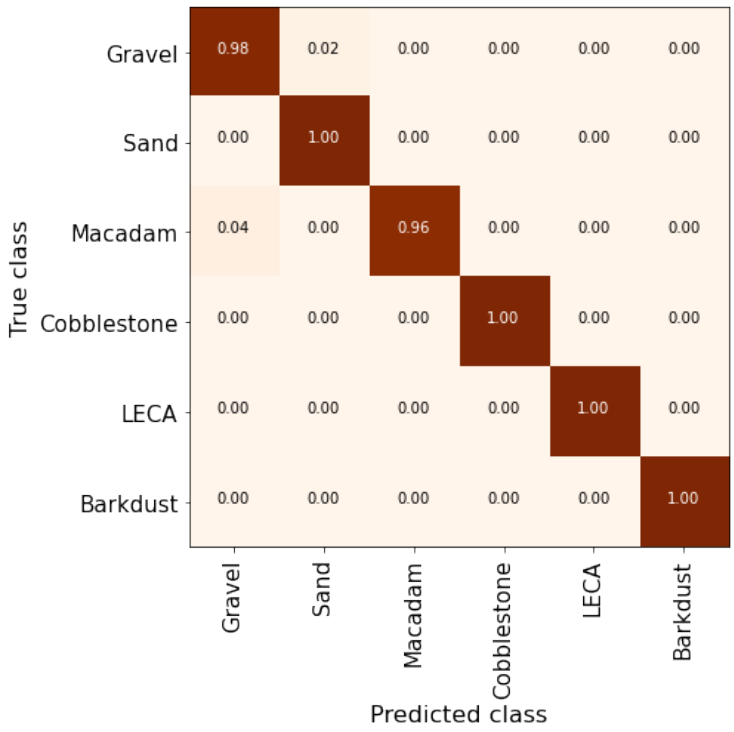


Figure 4.7: Confusion matrix with seven classes. Correct predictions lie on the diagonal.

The main observation in Figure 4.7 is that the large values are on the diagonal. Thus, the network is accurate in its categorized material predictions. There appears to be a small confusion between macadam and gravel, as well as gravel and sand.

5

Discussion

This chapter contains a discussion around the results from each campaign and the complete solution. The choice of hardware and placement is discussed. Finally, the developed system is considered from a larger perspective and future work is suggested.

5.1 Fill-Factor Estimations

The results from the initial test provide insight into the feasibility of the presented methodology. The static setup is believed to be optimal, since no occlusions from the container occur. Furthermore, fill-factor variations only affect the height, which is directly reflected in the distance found by the sensor. Hence, variations are fully observable from the sensor position. Performance of the initial static test is considered decent, since MAPE and MAPE_{max} are 5.1 % and 2.5 % respectively. This indicates that an optical load detection system is possible with the proposed setup.

Localization of the container in the static setup is expected to be good. This is due to the container not being moved between measurements, contrary to the excavator or wheel loader bucket. Bounding boxes are found to a high degree, both visually and numerically, as shown in the samples in Figure 4.1 and Table 4.1. Similar can be seen in both the excavator and wheel loader samples in Figures 4.2 and 4.3. The bounding boxes for the excavator and wheel loader are visually offset more from the ground truth boxes, compared to the initial static test. However, the overlap is high, as shown by the mean IOU in Table 4.1.

It is evident from Table 4.1 that the static test has the best performance throughout all metrics and campaigns. The relative errors for the excavator and wheel loader campaign are slightly worse. Fill-factor errors correlate to larger errors in absolute weight for the machines, since the quantity of material is considerably higher. Estimations are expected to be more difficult on the machines, due to the higher complexities of the setups.

From the error plots in Figures 4.4, 4.5 and 4.6, it is clear that the network has a biased prediction. For low weights, it tends to overestimate and for higher weights underestimate. The exception is the static test, where all predictions are overestimations. The excavator and wheel loader are usually loaded to a large fill-factor to

utilize their load capacity, and the most accurate predictions are provided in this interval. Moreover, the relative error decreases as the loaded weight increases. Thus, operations in the upper range of fill-factor (and weight) are optimal for the system. It is observed that data distribution plays a large role in the general appearance of the sample error plots. Ideally, data should be evenly spread across the entire interval to obtain a balanced dataset and avoid bias during training. If it was the case, the slope of the mean error line would become flatter.

Table 4.2 presents the sum of the sample errors for each campaign. These sums are interesting since high accuracy for individual measurements might not be required, but rather for a couple of measurements. When loading a truck or similar, the final weight is usually most important. The table shows that the system overestimates the weights, but not significantly. The relative errors are deemed excellent, especially for the excavator.

5.2 Material Classification

The developed CNN architecture appears to yield sufficient feature maps to identify the investigated set of materials. The test set is composed of images captured from the same material piles as the training images, but from altered perspectives. Hence, similar images appear in both training and evaluation, which may affect the credibility of the result. Evaluation using the test set reveals high classification accuracy, with few incorrect predictions, as can be seen in the confusion matrix in Figure 4.7. It is observed that the confused material classes tend to be of the same or similar types. The main difficulties in prediction appear to be between materials with similar appearances, such as classifying sand-like materials or determining fractions and sizes of gravel and macadam.

There are decisions to be made about materials of interest and which materials can be categorized into larger groups. It is expected that the type of the handled materials is not altered too much. For instance, the predictions could be limited to the expected set of available materials within a work site. Thus, the neural network could be solely trained on a smaller set of materials of interest.

5.3 Hardware

The purpose of the developed system was to provide a prototype. As such, the stereo camera is one choice out of many that can provide the necessary data for the methodology. The camera is deemed necessary for material classification. However, depth information can be gathered from a different sensor, which may provide higher resolution and more accurate data. The stereo camera used is a reasonable choice, since the methodology requires RGB images and depth data. It also provides built-in algorithms for calculating depth, resulting in a compact and convenient assembly, suitable for prototyping.

Together with the choice of sensor, placement is an important consideration. The sensor has to be out of the way of moving machinery and surrounding objects. The stated operating range of the stereo camera is potentially a limiting factor. For the wheel loader, the distance between the sensor and the bucket exceeds the recommended operating range set by Intel for accurate depth measurements. It is unclear how this affects the system's performance, and has to be considered in further developments. The sensor placement for the wheel loader is difficult, due to the bucket pose relative to the sensor. The optimal placement to not obstruct the machinery would be on the cabin. However, this would result in an occluded view into the bucket, limiting the amount of information that can be extracted with the depth sensor considerably. The constructed extension that holds the camera is pointing up and forward from the roof. This is not a practical implementation. It both inhibits movement of the arm of the wheel loader and may interfere with the surroundings.

Considering the excavator, the setup is not far from the scenario from the static test. The camera can capture the majority of the bucket with few occlusions. However, the bucket may be at different angles, and it is non-uniform compared to the static test. This may provide additional challenges for the fill-factor network. Placement on the stick is beneficial due to the distance from the bucket to the camera is similar throughout all movements of the excavator. Nevertheless, the camera has to be moved away from the boom-stick joint as there is a risk of crushing the sensor. If it is to be placed on the cabin, the distance to the bucket would vary widely. The consequence would be an additional challenge of handling variations in distances between measurements.

5.4 The Complete Solution

The implemented Faster R-CNN manages the fill-factor estimation and bucket localization jointly. Scales and ratios are tuned to match the sizes of objects of interest in the images. The objects are of similar size throughout all images, making it possible to tune the parameters to a small range. One inconvenience with the architecture is the classification layer at the end of the network. Since the desired value is continuous, a regression could be used in place of weighted sums. However, this requires modification to the existing framework.

Intervals of fill-factors and weighted sums are deemed as appropriate solutions to retrieve a continuous prediction. The intervals predicted by the network are determined based on the dataset. Tighter fill-factor intervals result in an increased number of classes. This increases the complexity of the network and consequently the amount of required data. However, a higher quantity of intervals potentially provides finer calculations of the continuous output. The Faster R-CNN architecture is used for the developed system and no other variants were investigated. However, the methodology is not limited to the chosen network architecture.

The reference data is obtained through on-board weighing systems, which have mea-

surement errors. The performance of the fill-factor estimation is directly influenced by the quantity and quality of the reference data. The evaluation does not take into account measurement errors of the on-board weighing system, since it is considered as ground truth. Further errors are caused by the weight and fill-factor conversion, as the density is assumed constant across measurements. The impact of this assumption is not entirely evident, since there are variations in the compactness and composition of the materials.

Using fill-factor, rather than weight or volume when training the network, has several benefits. Firstly, all measurements become independent of what type of material is loaded. The depth profile is similar regardless of what is in the bucket, given the same volume of material. Secondly, different bucket sizes with a similar depth profile can be evaluated with the same network. Thirdly, the reference data can be obtained through either volume or weight, depending on availability.

No investigation regarding environmental conditions, such as light and weather, was conducted. However, it is hypothesized that the conditions affect several parts of the system. Firstly, the density is affected, which may cause further prediction errors. For instance, the data acquisition for the excavator conducted over several sessions. Since the humidity in the gravel changes between the sessions, the assumption about constant density across measurements fails. Secondly, the quality of the RGB image is affected by light and weather, possibly obfuscating it. This may result in an incorrect material prediction. Contrary, fill-factor estimations are, in theory, more robust against changes in the scene. A requirement is that the sensor acquiring depth data is suited for the conditions. Dust, dirt and similar could disturb the stereo camera. The depth information is lost if one of the sensors in a stereo camera is occluded. Furthermore, using a stereo camera for depth sensing will not be possible in low-light conditions. The reason is that insufficient details for aligning the image pair are visible. Other types of sensors, such as a LIDAR or RADAR, may be more suitable in that case.

5.5 System Improvements and Future Work

The results of the fill-factor estimation and material classification demonstrate that the optical load weighing solution is viable. Nonetheless, there are improvements available for the solution. The developed system is constructed with a limited amount of information about the construction machine. Utilizing more information could prove beneficial. For example, in the work by Rasul et al. in [5] and presented in Section 1.2, sensor fusion of a stereo camera, a LIDAR and pressure sensors in the machine was performed. In a similar fashion, sensor fusion between an optical and an on-board weighing system can provide more accurate and robust predictions. Furthermore, pre-processing of depth data could take into account bucket pose. The pose could either be estimated through an optical system or measured by IMUs mounted on the machines. Additional information about the machine can be provided by the operator, such as bucket dimensions and sensor placement, to provide a more robust solution.

Being able to apply the trained network to various machines and bucket compositions would be desirable. Large quantities of data are imperative for training the networks. A suggestion is to continuously capture data with the sensor while operating the machines within the industry. In that way, a diverse data set can be established with variations in machines, buckets, environmental conditions and material types. The reference data should ideally be in terms of volume to bypass the influence of material properties when converting weight to fill-factor. Additionally, to achieve generalization, the distance interval is adjusted such that it reflects the fill-factor levels of the setup used for training. A criterion for this to be valid is that the bucket profile has to be similar, as the sensor has no way to detect what is happening below the material surface. The intervals used in the conducted experiments were tuned manually by visual inspections of the depth images. Since the intervals should be consistent for variations in machine models, bucket models and sensor placements, a set of calibration steps would be required. It could either be done manually, by examining reference depth data or by bucket alignment based on point clouds.

Considering the material classification, the operator can be notified and prompted to verify that the predicted material is correct. The network could take into account that the type of the loaded material will not be altered frequently in a session. Moreover, the material classification can consider location information, since it is expected that there is a limited set of materials within a site. The location could be tracked through a positioning system or provided by the operator. Combining material classifications with location information can be used for material tracking purposes. Future work can also be dedicated to evaluating the network with additional materials captured in various weather and light conditions. For certain materials, such as macadam and gravel, there is a possibility to apply and evaluate neural networks or threshold and segmentation algorithms for determining fractions.

When implementing the optical load weighing system in the machines, it can be presumed that detection is performed in real-time. A decision has to be made on how to predict the weights over a time series. For example, fusing multiple predictions over time, or a moving average of the estimation could be used. Knowledge of the pose of the bucket could be beneficial when capturing the data. This is to determine when the bucket is in an optimal position for the field of view of the sensor.

Applying the system to other machines is a possibility. The principle of determining fill-factor and weight for a bucket or container is similar regardless of shape. For instance, applying the proposed system on a truck or similar is deemed a possibility. Lastly, the sensor is not required to be mounted on the machine of interest. The optical load detection could be constructed as an external system, functioning similarly to scales placed around work sites.

6

Conclusion

An optical load detection system is developed to measure loaded weight in the bucket of two construction machines: an excavator and a wheel loader. The system relies on depth and RGB data for estimating the type, location, volume and weight of loaded material. The choice of hardware for acquiring the required data is a stereo camera. It captures depth data and RGB images simultaneously. Depth images are produced by filtering and mapping the depth data through a color map. For fill-factor estimation, a Faster R-CNN architecture takes the depth image as input and predicts fill-factors and a bounding box enclosing the material.

Three campaigns are conducted for three different setups. The first campaign utilizes, what is considered, the optimal setup for capturing depth information from the material. This campaign confirms the feasibility, and indicates the achievable performance of the system. The other campaigns correspond to the excavator and wheel loader. The sensor is placed on the stick on the excavator. This is deemed a good solution to get a sufficient field of view of the bucket. The wheel loader sensor placement requires consideration, as the current solution inhibits the movement of the bellcrank.

Evaluation reveals that the system can estimate fill-factor to a mean absolute percentage error (MAPE) relative to the maximum value of 3.3 % and 3.0 % for the excavator and the wheel loader respectively. Furthermore, the predicted bounding box is combined with the captured RGB image to filter out the background from the image. It is then fed through an additional network that predicts the type of material. The knowledge about material type is essential for providing density information to the weight calculation. The classifier is capable of identifying a range of construction materials to high accuracy, only confusing similar material types.

Future work should be dedicated to generalizing the system. For example, being able to use the system on additional machines, such as trucks and haulers, is a possibility. Other sensors and network architectures should also be investigated. Lastly, the system should be evaluated on a more diverse dataset, including various bucket sizes and weather conditions for example.

To conclude, the investigation reveals that an optical load weighing system is feasible given three criteria: a fill-factor estimation, a conversion between fill-factor and reference volume and the density of the material.

Bibliography

[1] E. Rundgren, "Automatic volume estimation of timber from multi-view stereo 3d reconstruction," M.S. thesis, Computer Vision Laboratory, Linköping University, Linköping, Sweden, 2017.

[2] P. Artaso and G. López-Nicolás, "Volume estimation of merchandise using multiple range cameras," *Measurement*, vol. 89, pp. 223–238, Jul. 2016.

[3] Y. Liang and J. Li, "Deep learning-based food calorie estimation method in dietary assessment," *ArXiv*, vol. abs/1706.04062, 2017.

[4] J. Lu, Z. Yao, Q. Bi, and X. Li, "A neural network–based approach for fill factor estimation and bucket detection on construction vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 12, pp. 1600–1618, Dec. 2021. DOI: 10.1111/mice.12675.

[5] A. Rasul, J. Seo, and A. Khajepour, "Development of integrative methodologies for effective excavation progress monitoring," *Sensors*, vol. 21, no. 2, p. 364, Jan. 2021. DOI: 10.3390/s21020364.

[6] R. B. Fisher and K. Konolige, "Range sensors," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 521–542. DOI: 10.1007/978-3-540-30301-5_23.

[7] P. Zanuttigh, G. Marin, C. D. Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, *Time-of-flight and structured light depth cameras technology and applications*. Springer International Publishing, 2018.

[8] M. Aboali, N. Abd Manap, A. Darsono, and Z. Yusof, "Review on three dimensional (3-d) acquisition and range imaging techniques," *International Journal of Applied Engineering Research*, vol. 12, pp. 2409–2421, Jun. 2017.

[9] S. Zhang, "High-speed 3d shape measurement with structured light methods: A review," *Optics and Lasers in Engineering*, vol. 106, pp. 119–131, 2018. DOI: https://doi.org/10.1016/j.optlaseng.2018.02.017.

[10] V. John, Q. Long, Y. XU, Z. Liu, and S. MITA, "Sensor fusion and registration of lidar and stereo camera without calibration objects," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, pp. 499–509, Feb. 2017. DOI: 10.1587/transfun.E100.A.499.

[11] R. I. Hartley and A. Zisserman, *Multiple view geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] J. Zhang, R. Du, and R. Gao, "Passive 3d reconstruction based on binocular vision," in *International Conference on Graphic and Image Processing (ICGIP 2018)*, vol. 11069, May 2019, p. 124. DOI: 10.1117/12.2524355.

[13] S. Mattoccia, "Stereo vision: Algorithms and applications," *University of Bologna*, vol. 22, 2013. [Online]. Available: http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf (visited on 04/13/2022).

[14] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, 2001, pp. 131–140. DOI: 10.1109/SMBV.2001.988771.

[15] K. Y. Kok and P. Rajendran, "A review on stereo vision algorithm: Challenges and solutions," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 13, pp. 112–128, Nov. 2019. DOI: 10.37936/ecti-cit.2019132.194324.

[16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[17] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv*, vol. abs/1511.08458, 2015.

[18] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *ArXiv*, vol. abs/1905.05055, 2019.

[19] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems*, A. Lotfi, H. Bouchachia, A. Gegov, C. Langensiepen, and M. McGinnity, Eds., Cham: Springer International Publishing, 2019, pp. 191–202.

[20] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14, Montreal, Canada: MIT Press, 2014, pp. 3320–3328.

[21] S. T. Krishna and H. K. Kalluri, "Deep learning and transfer learning approaches for image classification," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S4, pp. 427–432, 2019.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Nov. 2013. DOI: 10.1109/CVPR.2014.81.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015.

[25] Intel, *Intel realsense d400 series product family datasheet*. [Online]. Available: https://dev.intelrealsense.com/docs/intel-realsense-d400-series-product-family-datasheet (visited on 02/28/2022).

[26] Intel, *Depth camera d435i*, 2021. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/ (visited on 02/28/2022).

[27] Intel, *Intel realsense sdk 2.0*. [Online]. Available: https://www.intelrealsense.com/sdk-2/ (visited on 04/28/2022).

[28] Intel, *D400 series visual presets*. [Online]. Available: https://dev.intelrealsense.com/docs/d400-series-visual-presets (visited on 02/28/2022).

[29] *Colormaps in opencv*. [Online]. Available: https://docs.opencv.org/4.x/d3/d50/group__imgproc__colormap.html (visited on 02/28/2022).

A

Material Images, Densities and Full Confusion Matrix

(a) Gravel type 1 0-16 mm

(b) Gravel type 2 0-32 mm

(c) Gravel type 3 8-11 mm

(d) Gravel type 4 2-5 mm

(e) Gravel type 5 8-16 mm

(f) Gravel type 6 16-32 mm

(g) Gravel type 7 8-16 mm

(h) Gravel type 9 8-11 mm

(i) Barkdust type 1 20-50 mm

(j) Sand type 1 0-8 mm

(k) Sand type 2 0-2 mm

(l) Sand type 3 0-2 mm

(m) Cobblestone type 1 100-250 mm

(n) LECA type 1 12-20 mm

(o) Macadam type 1 16-22 mm

(p) Macadam type 2 8-16 mm

Figure A.1: The materials used for training and evaluating the material classification network. The input to the network is extracted patches from the center of the images.

Figure A.2: Gravel type 8 0-32 mm, used for training and evaluating the fill-factor network.

Table A.1: Material categories and approximate density intervals (tons per cubic meter).

Material Category	**Est. Density** $[t/m^3]$
Macadam	1.3-1.4
Barkdust	0.5-0.6
Gravel	1.2-1.5
LECA	0.3-0.4
Cobblestone	1.4-1.6
Sand	1.2-1.3

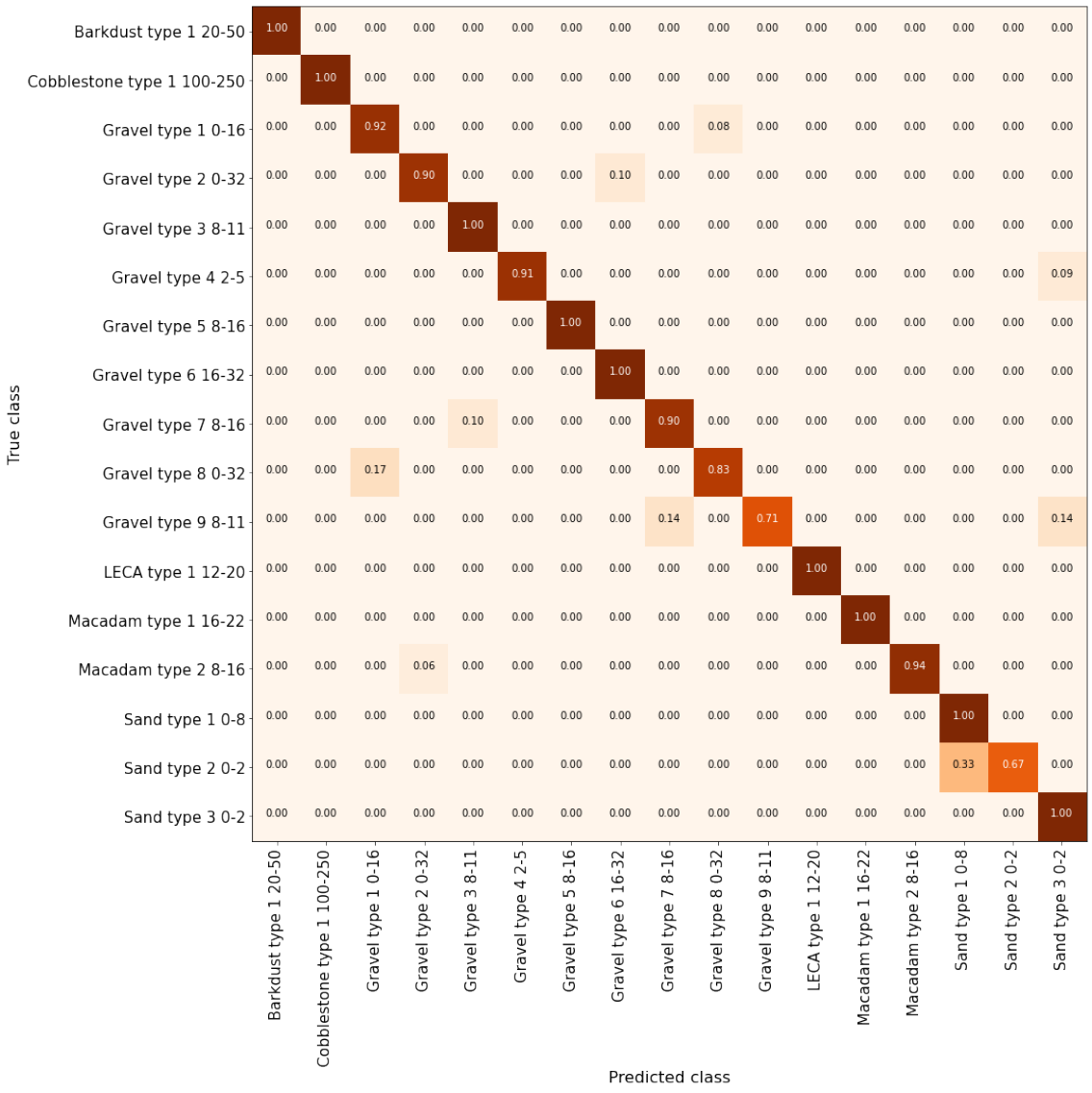


Figure A.3: Confusion matrix including all individual material types. Class names indicate the type of material and the fineness. For instance "Gravel Type 1 0-16" refers to one type of gravel in the dataset with grain sizes in the interval 0 mm to 16 mm.

DEPARTMENT OF VEHICLE ENGINEERING AND AUTONOMOUS SYSTEMS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY