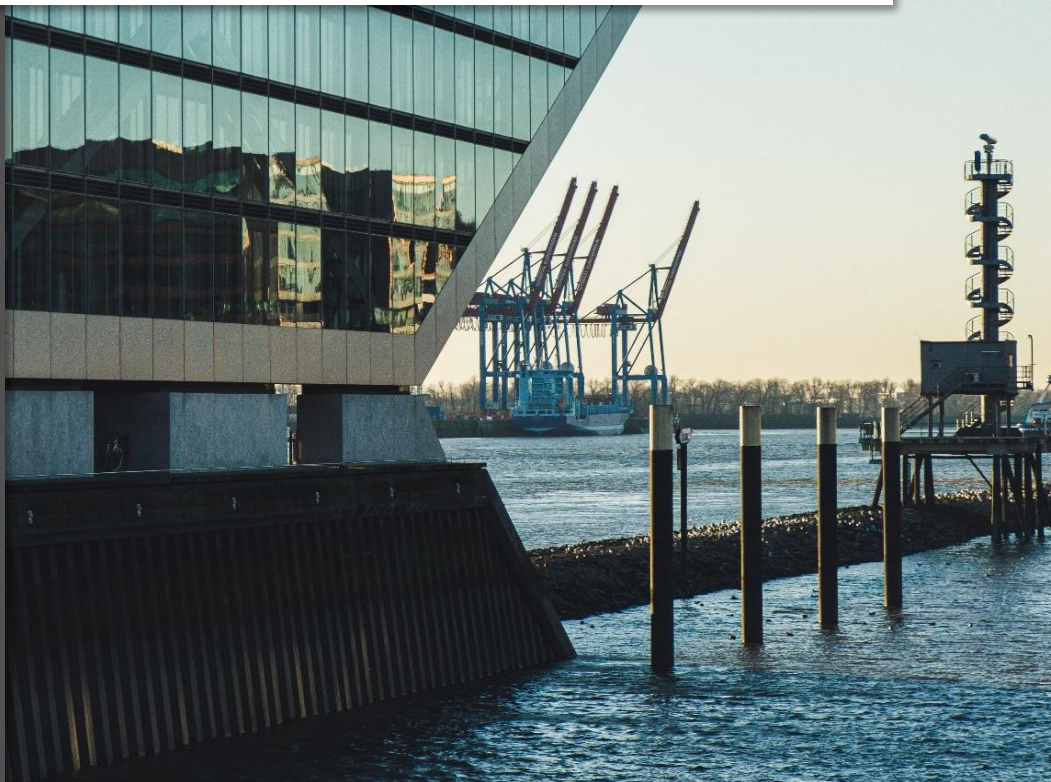Robert Brylka, Benjamin Bierwirth, and Ulrich Schwanecke

AI-based recognition of dangerous goods labels and metric package features

HICL

AI-based recognition of dangerous goods labels and metric package features

Robert Brylka[1], Benjamin Bierwirth[2] and Ulrich Schwanecke[1]

1 – Hochschule RheinMain

2 – Frankfurt University of Applied Sciences

Purpose: *Dangerous goods shipments require special labeling, which has to be checked manually every time a shipment is handed over in the supply chain. We describe an AI-based detection methodology to automate the recognition of dangerous goods labels and other shipment features (such as single piece volume detection).*

Methodology: *We use five industry RGB cameras and three AZURE RGBD cameras to generate images from shipments passing through a gate. The images are processed based on the YOLO detector to identify and separate dangerous goods labels and barcodes. We trained YOLO for our particular problem with about 1.000 manually labeled and 50.000 artificial generated images.*

Findings: *While dangerous goods labels detection was successfully validated in a laboratory environment and a warehouse, volume detection for single pieces consolidated on a pallet could be conceptualized. The system shows a high detection rate combined with fast processing, where the addition of computer-generated training images significantly improves the recognition rate for complex backgrounds.*

Originality: *Parallel detection of multiple package features (volume, barcode, dangerous goods labels) of multiple pieces consolidated on a pallet is not available yet. Our solution processes a shipment faster and more accurately than existing single-piece solutions without restrictions to the material flow.*

AI-based recognition of dangerous goods labels and metric package features

1   Introduction

Transportation planning and handling are based on information about the packages shipped. As the various modes of transport or warehouses have limitations related to weight and volume, these parameters are of high interest to optimize consolidation and transport efficiency. Unfortunately, in practice, data provided by the shipper is not very accurate. Therefore, in many industries measuring weight and volume at a logistics service provider location has become standard. The corresponding processes are time-consuming and require special equipment to measure the dimensions of a shipment. While shipments in courier, express, and parcel logistics are processed as individual pieces in automated handling and sorting systems, skid-based handling takes place in general cargo logistics. As a shipment consisting of multiple individual packages (pieces) is handled with one or more skids, the dimensions of those skids are measured instead of the dimensions of the individual pieces. The results are higher total volumes than the actual volume. Especially for modes of transport where volume is a significant limitation, e.g., in air transport, this leads to inefficient transport planning as transport volume could remain unused because the planning algorithm and its solution were based on exaggerated shipment volumes. Transforming the handling processes from skid-based handling to piece-level handling would cause inefficiencies in the handling processes (as the reason behind the invention and use of loading devices like skids was the idea of increased efficiency based on consolidated handling of multiple pieces). Therefore, solutions for measuring single-piece dimensions are required.

Special handling requirements further impact transport planning and handling. One of the most critical aspects is information about dangerous goods. In case of an accident or other emergency, these goods pose a safety risk as they could explode, ignite, intensify a fire or cause harm to nature. The various types of dangerous goods are classified into several classes (Nations, 2019). These items have to be handled with care. To limit the risks in terms of the impact (and not of the probability) of an incident, the quantities that could be stored or transported together are limited and related to other requirements, such as additional fire protection equipment. There are also co-loading restrictions which means that only limited quantities of certain dangerous goods could be stored or

transported together. To ensure correct handling and thereby reduce the probability of an incident, pieces containing dangerous goods must be clearly labeled and thoroughly checked at acceptance and when leaving a warehouse or other handling facility. The checking procedures are typically initiated by the accompanying information flow or paperwork. The checking of procedures consists of checking the correct labeling of the relevant pieces, more thorough checking of damages to the package and outgoing shipments, the clear visibility of those labels (to simplify identifying critical pieces in case of emergency). The checks and the transport require special training and cannot be done by every logistics employee. Shipments could remain undetected even if they are correctly labeled if the corresponding shipping documents do not indicate dangerous goods. To further minimize the risks and accelerate the check-in and check-out processes of dangerous goods, assisting technology should be assessed.

In this work, we present a fully automated approach for detecting and analyzing various characteristics of multi-piece shipments on a skid or pallet. Our aim is to identify and decode barcodes and dangerous goods labels (DG labels) without interfering with the material flow. Additionally, we describe an approach to measure the volume of those pieces while being stacked on the skid. We first describe the relevant related work. Then, we present our pipeline to detect and decode barcodes and DG labels in the method section. In the same section, we present the approach for the automatic generation of training data sets for volume measurement. Subsequently, we discuss the results that we have achieved with our approaches. Finally, we give a short conclusion and discuss possible future work.

2   Literature Review

In this section, we provide a brief overview of relevant and recent publications. We first review the relevant publications on the camera-based acquisition of 2D package information, such as barcodes and DG labels recognition. Second, we discuss the current situation and related work on package volume estimation.

2.1 Barcodes and DG labels recognition

A large number of publications about barcode localization and decoding exist. The classical methods mostly use low-level image features as gradient analysis, build of saliency maps, Hough space analysis, or probabilistic methods (Gallo and Manduchi, 2011; Sörös and Flörkemeier, 2013; Creusot and Munawar, 2015; Namane and Arezki, 2017). Some of these approaches have further geometric restrictions, e.g., that the barcode must be horizontally aligned (Gallo and Manduchi, 2011) so that the algorithm is not suitable for detecting barcodes on real, freely arranged goods. Other approaches are unable to deal with blurred or poorly lit images (Creusot and Munawar, 2015; Namane and Arezki, 2017). However, all of them have a high computational effort in common, which, combined with the high resolution of today's cameras, makes them not real-time capable. For over a decade now, neural networks have been successfully used for barcode localization (Zamberletti, Gallo and Albertini, 2013; Hansen et al., 2017; Zhang et al., 2018). In Zamberletti (Gallo and Albertini, 2013), the authors use a Hough transform and a simple Multi-Layer-Perceptron to achieve good results if the barcode is very present in the image. Hansen et al. (2017) detect barcodes utilizing the YOLO Detector (Redmon et al., 2015). This approach shows good detection performance, but it does not accomplish accurate segmentation of the barcodes. Zhang et al. (2018) presented another localization approach based on deep learning but used special acquisition hardware to keep the input images well illuminated and nearly noise-free. Recently Brylka (Schwanecke and Bierwirth, 2020) presented an approach that covers the localization and decoding of barcodes under challenging circumstances. Their solution is explicitly aimed at the industrial capture of barcode information and focuses on correcting poorly conditioned images (poor illumination, meager barcode resolution, blurring). Our approach for barcode detection strongly relies on this approach and complements it with the DG labels detection methodology.

Many publications are primarily concerned with the detection and recognition of DG labels. For example, Pozo et al. (2013) showed two approaches based on edge detection and saliency maps. However, both methods only work if the label shows enough contrast to the background and has sharp contour. Also, the final categorization of the label is not fulfilled with these approaches. Mohamed (Tünnermann and Mertsching, 2018)

presented an approach based on spectral residual saliency (Lienhart, Kuranov and Pisarevsky, 2003) and a Scale Invariant Feature Transform (SIFT (Lowe, 2004)). Building the saliency map over the input image, they first localize the DG label and then search for the best label match in a SIFT descriptor database built from a training set of DG labels. This approach shows good results, at least if the label background is sufficiently different, but it is very slow. Also, many techniques based on Deep Learning have been presented in the field of DG label recognition (Edlinger, Zauner and Zauner, 2019; Sharifi, Zibaei and Rezaei, 2020; Tijtgat, Volckaert and Turck, 2017).

In particular, the YOLO detection system was often utilized. For example, Tijtgat (Volckaert and Turck, 2017) used YOLO to detect dangerous objects in images from a drone's camera. Using the positioning of the drone and the localization of the detected object in the camera image, they were able to approximate the GPS localization of the dangerous object. However, the authors focus on the overall system and limit the detection of DG labels only to one specific label selected for their purpose. Edlinger (Zauner and Zauner, 2019) used the YOLO detector for detecting multiple DG labels. However, the dataset used by the authors shows labels always on the same background and at a large distance from each other. This leads to limitations in the real scenarios, with different backgrounds and multiple labels at the same tiny package. The approach of Sharifi (Zibaei and Rezaei, 2020) is another application of YOLO as a detector of DG labels. The dataset used in this application contains images taken with different backgrounds, sometimes with poor lighting conditions and overlap. However, the DG labels are very present in the images. They lack the typical noise and motion blur in images of moving objects, so they are very different from real-world images for a logistic transportation scenario. We also utilize YOLO for detecting and decoding DG labels. However, we put the most significant weight on the easy adoption of the system to new situations, as the expansion of the DG labels set or the change of the working space. Therefore, one of our main focuses is the development of a training data generator.

2.2   Volume estimation

There are various approaches to determine the volume of an object represented by a point cloud. One method is to calculate the convex hull for the point cloud (Preparata

and Hong, 1977). However, this approach has several disadvantages. First, most objects are not convex, so the calculated volume will be significantly larger. Second, if the requested object is not fully represented, the volume will be much smaller. Furthermore, a typical load consists of several packages, which makes the assignment of the volume to each individual package challenging. Another approach is to approximate the sought object by primitives, such as boxes, cylinders, or spheres. However, methods that follow this approach (Goron et al., 2012; Figueiredo et al., 2019) only work well if the objects to be detected (single pieces) lie on a homogeneous, planar surface. A condition that cannot be guaranteed in a warehouse environment with forklifts, trucks, and people in motion.

In practice, there exist three different processes to measure volume. One is that the pieces are separated and measured piece by piece. This is typically done in the CEP (Courier, Express, Parcel)-Industry with the help of automated material handling and sorting systems. Here, the typical restrictions in weight and volume of pieces apply (e.g., Max. 36 kg) – so measuring a skid-sized box is not possible. The second option, which is more common in general logistics, is measuring the maximum dimensions of the shipment on the skid or pallet by creating a box that contains the full point cloud. A prerequisite of this approach is that one pallet carries only one shipment. The existing systems require user interaction for the identification of the shipment, typically by scanning the barcode. Based on this information, the measured volume is then saved in the warehouse management system. Therefore, measuring the volume is an extra process step that disrupts the material flow and takes extra warehouse space and labor time. The third option is to measure shipments manually, which is done for oversized cargo or if only a small percentage of all shipments are measured (Borstell, 2018).

For quite some time now, machine learning has also found its way into processing and information extraction from point clouds. Especially in autonomous driving, where point clouds generated by a LIDAR system are processed by deep learning methods to locate other road users, such as other cars, motorcyclists, bicyclists, or pedestrians (Fang et al., 2021; Lang et al., 2019; Shi, Wang and Li, 2019). However, for the autonomous vehicle, only the situation in the 'plane' of the surrounding roads is relevant. Therefore, in most approaches, a bird's eye view is chosen, and the problem is reduced to 2D localization. In the street situation, where road users are usually not on top of each other, this does not

lead to problems. However, if one wants to localize individual packages lying on top of each other, the lack of altitude leads to difficulties. However, some approaches deal with the localization and segmentation of objects away from the road. The S3DIS and ScanNet datasets (Armeni et al., 2016; Dai et al., 2017), for example, provide large-scale scans of office building interiors. These datasets are used to test the localization and segmentation of individual objects such as tables, chairs, walls, etc. (Pham et al., 2019; Yang et al., 2019). Most deep learning approaches realize supervised learning. In this case, a large number of annotated datasets is used for training deep neural networks. Since, to the best of our knowledge, there is no dataset that represents the annotated packages in a logistics process, in this project, we present an approach for creating such a dataset with minimal manual effort. With this dataset, we can test the transferability of existing approaches to our particular domain and validate new approaches more quickly.

2.3   Dangerous goods regulation

Transport and handling of dangerous goods are regulated based on global standards of the UN (Nations, 2019). There are different guidelines for the various modes of transport as the potential impact of an incident varies. For specific infrastructure an elevated regulations apply – for example, additional limitations have to be considered if a road transport passes certain tunnels. The strongest requirements apply in air cargo as any incident in the cargo compartment while in the air leads to a total loss of the airplane and all persons and the cargo on board. Several incidents related to lithium batteries led to additional restrictions and a new lithium battery mark (section 5.2.1.9 of the UN model regulation).

In general, the regulation defines nine different classes of dangerous goods. For those classes, requirements for packaging, labeling, handling, storage, and transport apply. Additionally, all personnel involved in the transport and handling process needs specific training to ensure compliance.

Special focus in the transport chain is, of course, on the handover of dangerous goods shipments. For incoming and outgoing shipments correct labeling (for all labels, see section 5.2.2.2.2 of the UN model regulation) and integrity must be checked thoroughly (Part 5 and Part 7 of the UN model regulation).

AI-based recognition of dangerous goods labels and metric package features

Correct labeling ensures that the critical pieces could be identified immediately in case of an incident, and the response from the emergency services could be adapted accordingly. Ellis (2010) showed that up to 7% of wrong labeling was discovered by authorities in maritime shipping.

3 Method

In this section, we discuss the hardware configuration of our camera system, which was developed specifically for this purpose. The description of our software system can be divided into two main aspects: the detection and decoding of 2D information and the 3D volume determination of the load. In the following, we will discuss these three topics separately.

3.1 Camera setup

The main challenges in capturing the visual characteristics of air cargo shipments are:

- the large scanning area of about 2.5 x 2.5 x 2.5 meters
- insufficient light conditions
- high speed at which the cargo can be transported with a forklift

To meet this challenge from a technical point of view, we developed a camera setup whose schematic structure can be seen in Figure 1. Our camera system consists of five high-resolution industrial RGB cameras and three Azure Kinect RGB-D cameras. All cameras are hardware synchronized and are connected to a central computer.
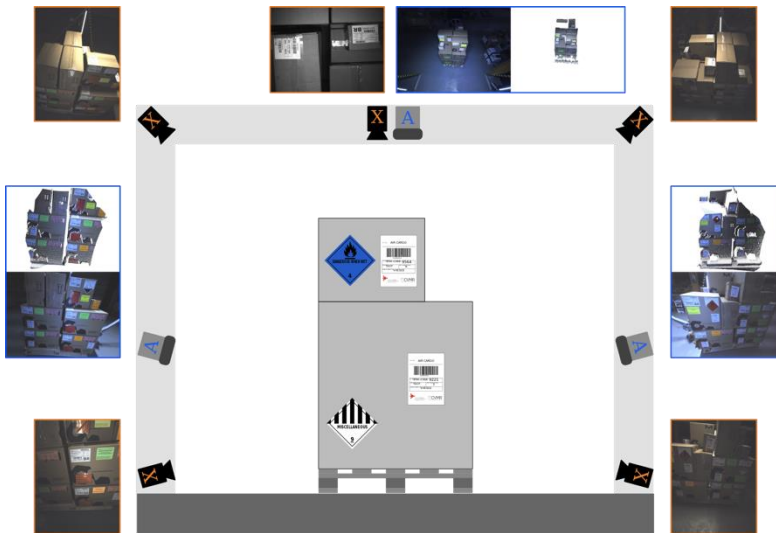
Figure 1: Shipment handling camera setup

Our system consists of five high-resolution cameras (marked with orange X) and three Azure 3D scanners (marked with blue A) and enables extensive coverage of the five (visible) sides of the cargo.

The industrial cameras all have a resolution of 12.4 MP, are equipped with a global shutter, and can acquire images in RAW format. These features make these cameras primarily used for capturing and decoding barcodes and DG labels.

The Azure 3D scanners work with the so-called Time-of-Flight method, enabling the creation of depth maps that we use to measure the volume of packages. In addition to depth measurement, any Azure device has a 12.4 MP color camera sensor. These cameras are equipped with rolling shutters and provide the captured images with JPEG compression, limiting their use to detect barcodes or DG labels.
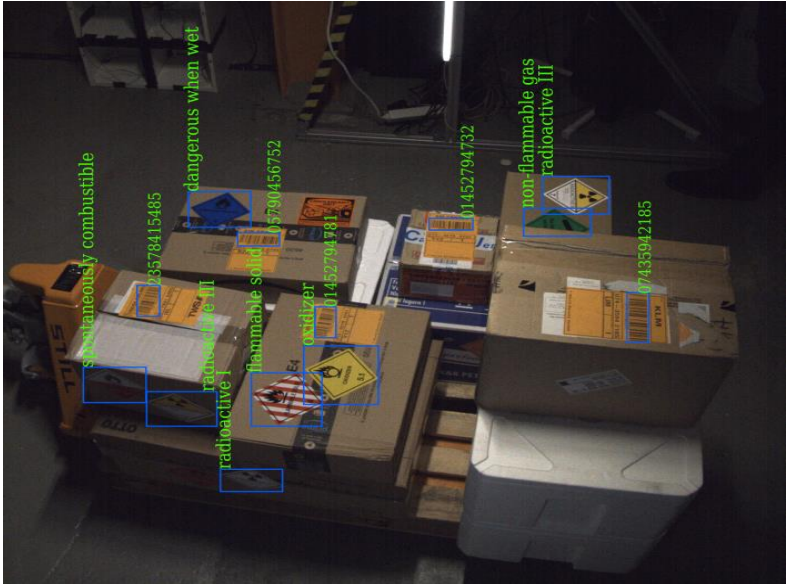
Figure 2: Exemplary presentation of the barcodes and DG labels detection and decoding

3.2 Barcodes and DG labels

The 2D detection of our system focuses on the barcodes and DG labels. As shown in Figure 2, we like to detect and decode all the existing relevant barcodes and DG labels for each individual input image.

We limit the selection of relevant barcodes to those provided on the Air Way Bill (AWB) labels. A template of an AWB is shown in Figure 3 left. The design, in the form of a rectangle with a dominant barcode and additional information, is specified by a standard. We limited the selection of DG labels to seventeen pieces for our test scenario, which are essential in air cargo (see Figure 3 right).

Figure 3: 2D information labels. Left: AWB template in white, right: selection of seventeen DG labels used in our scenario

Barcodes: As indicated in related work, we have followed the work of Brylka (Schwanecke and Bierwirth, 2020) to detect and recognize the barcode information. Figure 4 show the main steps of the barcode detection and decoding pipeline:

- coarse localization of the barcode
- limiting the barcode area to be considered based on this localization
- refinement of the localization by segmentation of the barcode's stripes
- determination of the orientation and narrowing the area to a quadrilateral
- horizontal alignment and mapping to a rectangle
- deblurring of the barcode and subsequent decoding

First, bounding boxes of the barcodes are determined using the real-time detection system YOLO (Redmon et al., 2015). Based on this localization, a more refined segmentation of the barcode bars and the determination of their contours is performed. This step is performed with an additional neural network U-Net (Ronneberger, Fischer and Brox, 2015). In this case, the segmented foreground represents the barcode stripes and classifies all the surroundings as a background. Relying on this segmentation, the orientation of the barcode is determined, and the final square contour is determined (see cyan contour, with a solid line representing the barcode base in Figure 4). With the specified orientation, we can align the barcode horizontally, or with the square contour, we can rectify the barcode to a square. A nearly horizontal orientation of the barcodes is a precondition for decoding the barcodes with a standard barcode decoding library. In our case, we use the freely available image processing library ZXing (David Gilbert and Huy Cuong Nguyen, 2021). Since the images are generally very noisy and very dark, and blurred due to insufficient lighting, decoding of the barcodes very often fails at this point. This is where the strength of the algorithm (Brylka, Schwanecke and Bierwirth, 2020) comes into play. With the help of an additional U-Net network, the noisy, blurred barcode

bars are reconstructed into sharp ones. Subsequent decoding is performed using the ZXing library. The use of the standard barcode library takes over the position of the validation tool. Thus, the checksum verification finally determines whether the considered image section represented a barcode.
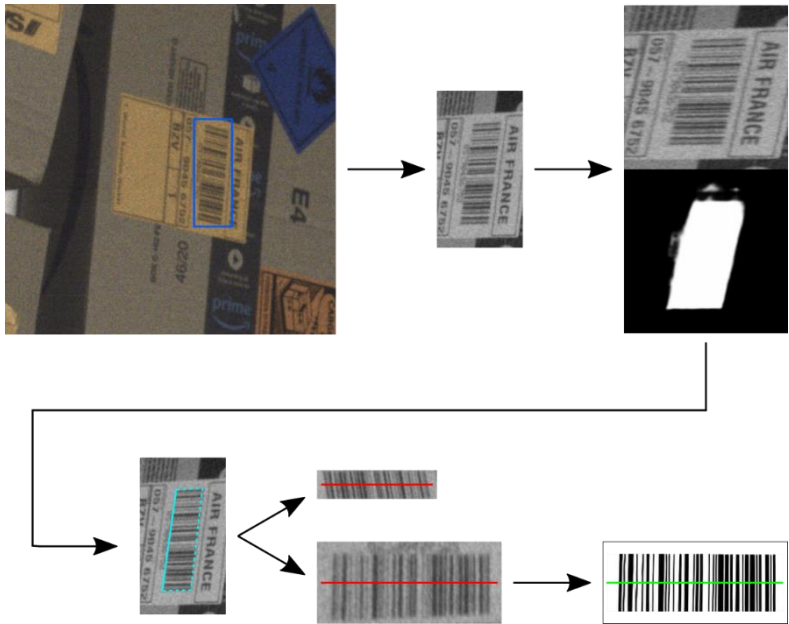

Figure 4: Main steps of the barcode detection and decoding pipeline

Following the path: coarse localization of the barcode, restriction to the relevant area, barcode bars segmentation, orientation and quadrilateral area estimation, horizontal alignment and rectangle mapping, deblurring, and decoding.

DG labels: Our detection and decoding of DG labels are mainly based on the YOLO detector. The detection with YOLO only leads to good results if there is a lot of labeled data in the training phase that covers all eventualities. If there is a significant deviation in the images in the test phase compared to the training data set, the results are likely to

be significantly worse. Adapting the system to the new conditions can be easily accomplished with additional training data reflecting the new situation. However, the manual labeling of the data is very dull, time-consuming, and often error-prone. For this reason, an essential part of the software component in the part of the DG labels detection was dedicated to the development of a training data generator.

Before meeting the requirements for the generator tool, we analyzed several records of real transports. In doing so, we found inconsistencies in the appearance of individual DG labels. The selection of DG labels we chose for our test scenario (see Figure 3) shows black lettering consistently. However, it is quite common for white lettering to be used as well. Like, for example, the label *liquid gas* in the reference Figure 5 on the left. Another observation shows the problem in the means of transport medium itself. As seen in Figure 5, center and right, forklifts can be labeled with many information stickers. These high-contrast labels, which are often very similar in shape and color to the real DG labels, can be distracting if they were not present in the training set. The goods themselves can also vary greatly, whether they are a wooden crate or pure cardboard, for example. Finally, the background itself can also vary. So, whether it is a truck loading area or if the camera system stands in the hall and the background is a random warehouse environment.

Figure 5: Challenging situations when recognizing DG labels

On the left: deviation from the standard white lettering, center and right: warning and information labels, which can lead to falsification of the result due to their similarity.

Finally, we implemented the following requirements in our generator tool:

- freely selectable background images (track bed, warehouse, etc.)
- free definition of the load (cartons, plastic boxes, wooden boxes, etc.)
- easy extension of the existing DG labels (color of the label, generally more characters, etc.)
- possibility of adding 'structured noise' (such as high-contrast labels, etc.)

Figure 6 shows some examples from the training dataset generated with our tool. For example, various product presentations can be seen, such as black plastic containers, fabric tarpaulins, or cardboard boxes. In addition, several DG labels and contrasting clutter objects can be seen on the load.

Figure 6: Example of training data created with our generator tool

To see are several DG labels projected on different goods. Also, to see are the disruptive elements in the form of high-contrast labels.

Volume estimation: As mentioned in the related work section, most of the current methods for detecting objects in 3D point clouds are based on machine learning. Thereby, supervised learning is used, which always requires labeled training datasets. We will present a method, which creates the annotation for the training datasets in a fully automatic way. As explained in the camera setup section, the Azure 3D scanners used in our setup provide a 3D point cloud and a color 4K image of the reconstructed scene. In our approach, we first localize the packages in the colored 2D image and then transfer the mapping to the reconstructed 3D point cloud.

One of the common ways to localize an object in an image is the use of fiducial markers. A fiducial marker is a separate element that is attached to the object to be identified. A widespread form of fiducial markers is the ArUco marker (Garrido-Jurado et al., 2015; Romero-Ramirez, Muñoz-Salinas and Medina-Carnicer, 2018). An ArUco marker is a plain 2D imprint composed of black and white rectangles (see Figure 7 left). With the freely available ArUco library, such markers can be easily localized in an image. Given the size of the marker and the calibration of the camera, the 3D pose (position and orientation) of the marker relative to the camera can be determined (Garrido-Jurado et al., 2015; Romero-Ramirez, Muñoz-Salinas and Medina-Carnicer, 2018).
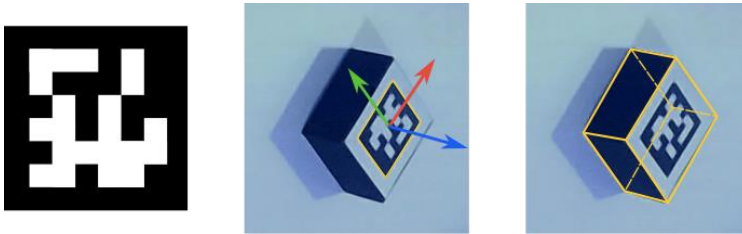

Figure 7: Fiducial markers; left: template example of an ArUco marker; middle: localization of the marker, right: bounding box for the detected object (yellow)

In our case, we use the 3D pose of the marker to determine the 3D contour of the relevant package and thus simultaneously mark the relevant points in the point cloud. For this purpose, we provide a selection of cardboard boxes with the ArUco markers on them. Figure 8 (left) shows an example of our prepared package. The three separate views are given by the three Azure cameras in our setup. As can be seen, each of our packing boxes has a total of 6 markers applied to the six possible surfaces. With this number of markers per box, it can be assured that detection from different view angles (cameras) is possible. Furthermore, the problem of overlapping by other packages can be reduced. As illustrated in Figure 8 (middle), we bring the three partial point clouds together. Therefore, we use the stereo calibration procedure (Zhang, 1999), which computes the position and orientation of the sensors to each other. As a result, we get a jointed point

cloud and the estimated bounding boxes registered to each other (cyan-colored boxes). Performing a Principal Component Analysis (PCA) over the collection of the bounding boxes, we estimate the final average bounding box, which represents the best enclosure box (Figure 8 right).
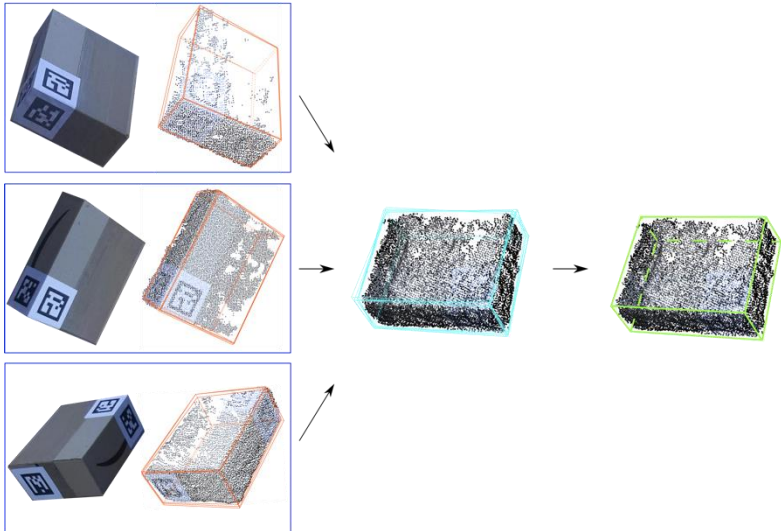

Figure 8: Automatic piece annotation

From left to right: three unregistered inputs (image and point cloud) from the Azure devices, registered point clouds, and registered bounding boxes given by ArUco library, last row: PCA resulting bounding box and final enclosed box point cloud.

With this method, a set of training datasets can be generated easily and used in deep learning approaches for object segmentation and detection in point clouds. Figure 9 shows an example of our dataset, with the registered point clouds from the three 3D scanners, the average enclosing boxes, and the annotation denoted by different colors (white marks whole irrelevant region).
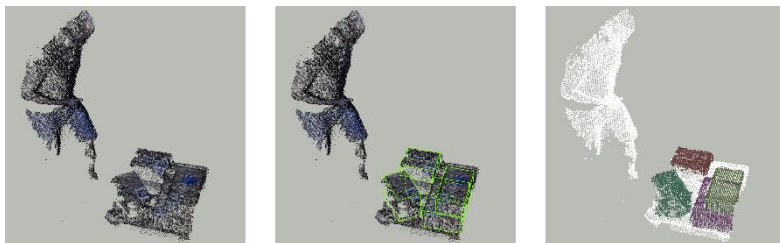
Figure 9: Preparation of real dataset: input point cloud, automatically detected enclosing boxes, annotation colorized according to different instances

4   Results

In this section, we present the results obtained by our approach. First, we discuss the results for the detection and decoding of 2D information. Second, we show the results of our approach for 3D volume segmentation using our automatic annotated dataset.

4.1   Barcodes and DG labels

In our barcodes and DG labels detection pipeline, a total number of three networks are used. The two U-Net networks, used for barcode contour segmentation and barcode deblurring, are trained with the exact settings as in the initial publication. For more details, please refer to Brylka (Schwanecke and Bierwirth, 2020). We tailored the YOLO detector to our scenario. In addition to the localization of barcodes, it also localizes the 17 different DG labels. We use a mixture of about 1.000 manually annotated data and 50.000 artificially generated data as a training set. The training set contains about 60.000 barcodes and about 9.500 of each DG label, distributed over all images. Because it has an enormous performance increase with a small loss in detection rate, we used the current YOLOv4-tiny version. We set the input size of the network to be 640 x 480, which is around one-sixth of the original image size (all cameras deliver images in 12.4 MP). This input size is a compromise between the performance and accuracy of the network and was

determined experimentally. We trained our model with batch size 64 with an initial learning rate of 0.001. After 50 epochs, there was no significant oscillation of the loss, which is also our stopping criterion.

Barcodes: We have evaluated the performance of our pipeline related to the localization and decoding of barcodes on the challenging dataset of Brylka (Schwanecke and Bierwirth, 2020). This dataset shows the transportation of cargo shipment under real conditions. This dataset has the ground truth annotation for a total number of 840 AWB's distributed over 400 images. There are no DG labels in this dataset. We achieved a 5% higher Recall than Brylka (Schwanecke and Bierwirth, 2020)on this dataset. This is mainly because our network input size of 640x480 is significantly larger than the original input size of 416x416. We also use real, manually labeled images in the training set. The mixture of real and artificial training data seems to perform better than just using artificial images. Finally, we used the latest, fourth YOLO architecture.

DG labels: We have created a new validation dataset for the evaluation of the detection of DG labels. This dataset consists of eight sequences showing the loading process of the cargo. We manually create ground truth annotations for the total number of 2.216 images and 5.820 DG labels. As described in the method section, the detection of DG labels is only based on the YOLO detector. Thus, the balance between the detection rate and the error rate can only be tweaked by the score threshold of the YOLO detector, i.e., at what score a detection is considered valid. In general, the higher the threshold, the fewer false detections. However, this also means that some of the DG labels can be overlooked.

Since the goods are screened several times as they pass under the camera arch, the detection rate is defined in practice as the number of recognized labels within a sequence. This means that the recognition rate is 100% if all DG labels are correctly recognized at least once in a sequence. Of course, to increase the reliability, the system should decode as much as possible of all existing labels per image. At the same time, however, the number of incorrect assignments should be kept small. We tested our system considering these aspects by using different values of the score threshold. Table 1 shows the obtained results with the Recall, Precision, and Detection Rate per Sequence metrics. The Recall is defined as the quotient between the correctly detected and all given DG labels. The system quality is measure by Precision which is defined as

the quotient between the number of correctly decoded candidates and all candidates suggested by the system for a given label. The Detection Rate per Sequence is defined as described above; a unique occurrence of a DG label in a sequence needs to be correct decoded at least once to get the full score. Table 1 summarizes the results over all seventeen DG labels and sequences.

As can be seen, our system can operate at the highest threshold setting without a single false detection. It achieves a Detection rate per Sequence of almost 80 %. The Recall is at a moderate 20 %. The low recognition rate per frame can be explained by the fact that the imaging of the DG labels varies significantly during the sequence. Therefore, at the entrance and exit of the scan area, the DG labels are very small (due to the angle of view of the cameras). So, it can be expected that only the DG labels in focus will get a high score, and thus positively affect the Detection rate per Sequence.

Table 1: Results of the DG labels detection over the validation dataset

YOLO Score Threshold	Recall	Precision	Detection rate per Sequence
99	0.197	**1.000**	0.795
95	0.286	0.995	0.885
90	0.325	0.990	0.885
85	0.351	0.987	0.923
80	0.368	0.982	0.949
75	0.385	0.976	**0.962**
70	0.395	0.971	0.962

YOLO Score Threshold	Recall	Precision	Detection rate per Sequence
65	0.407	0.968	0.962
60	0.418	0.961	0.962
55	0.427	0.955	0.962
50	**0.439**	0.951	0.962

When the threshold decreases, a positive trend in Recall and Detection rate per Sequence can be seen as well as a degradation in Precision. For a threshold value lower than 75 %, there is no change in the Detection rate per Sequence. This is because some of the DG labels are covered by other packets to such an extent that they do not contain enough information for the system. Based on these observations, we have determined the threshold value of 75 % as the optimum for the system. Further analysis revealed that many misclassifications fell into the classes 'flammable liquid' and 'flammable gas' as well as 'misleading' and 'c9a'. This can be explained by the fact that the labels of each of the two classes are very similar.

4.2   Volume estimation

We tested our approach to automatically generate training sets for point cloud segmentation on BoNet (Yang et al., 2019). The output of BoNet is a semantic segmentation, i.e., the assignment of points to a specific category as well as an instance segmentation. In instance segmentation, the points with the same semantics are assigned to individual instances. Initially, BoNet was tested on ScanNet and S3DIS (Dai et al., 2017; Armeni et al., 2016). These two datasets show the facilities of a typical office environment as well as large conference rooms. As described in the method section, our automatic annotation method splits the point cloud into two classes: packets and the rest. Each of the annotated packets is mapped to one particular instance.

As training data for BoNet, we automatically annotate packages on 150 pallets, which are always brown boxes with ten different dimensions (see Figure 9). For training BoNet, we used the default settings, a learning rate of 0.0005, and a batch size of 4. As a termination criterion, we restricted the maximum number of learning epochs to 300. To validate the results, we collected another dataset containing packages without the ArUco annotation and differentiating in size and color from the scheme of the training set. Figure 10 shows the results for part of the validation set. The first column shows the pallet arrangements. In the second column, we see the registered point cloud acquired with the three Azure scanners, which is the input to BoNet. The third column shows the semantic segmentation results, with cyan marking the points that belong to the package and white marking the background. The last column shows the instance segmentation results. The background is again marked in white. The different instances of the packages are marked with different colors. As can be seen, the semantic segmentation between the package instances and the background is very good.

AI-based recognition of dangerous goods labels and metric package features
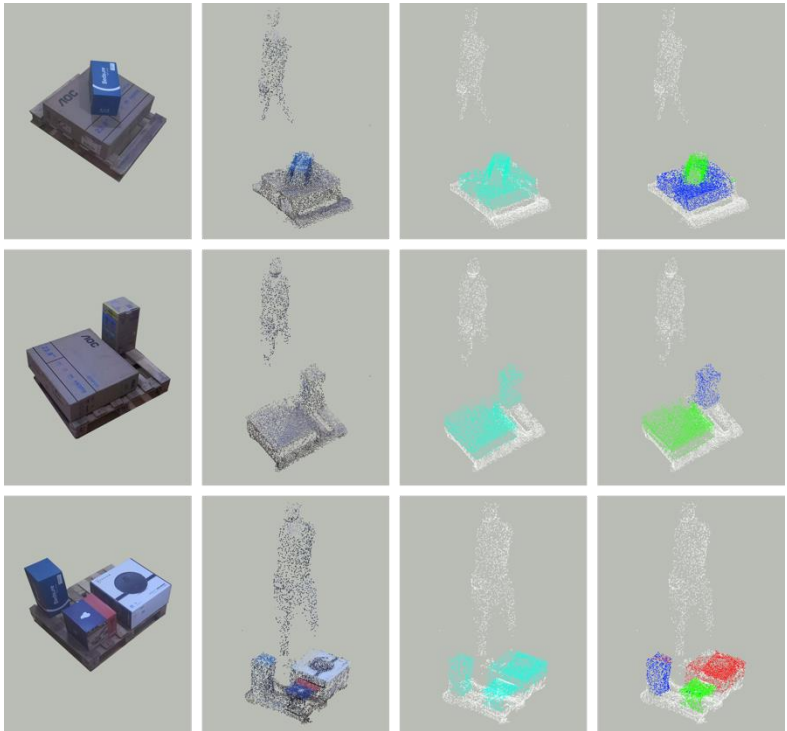

Figure 10: Visualization of the semantic and instance segmentation results

From left to right: Image of the scene, colored point cloud, segmentation of package and background, segmentation of individual packages and background.

Therefore, it is possible to isolate the packages and separate them from the background. The instance segmentation results show slightly more noise and imprecision. Nevertheless, points belonging to a specific package can be separated very easily. Subsequently, we can determine the smallest enclosing box for each separated point cloud. And thus, we can approximate the volume of a single package.

5   Discussion and outlook

The presented fully automated process enables significantly faster and safer handling of dangerous goods. Correct labeling could be automatically verified by matching against the corresponding information flow or paperwork. So far, only the integrity check would remain manual. Besides the handling of known and booked dangerous goods shipments, unidentified shipments (because the information and/or paperwork does not indicate dangerous goods contents, no explicit checking and handling would occur) would be identified and handled correctly. Additional to the faster checking and higher quality, the camera system would directly provide pictures to document deficiencies (e.g., damages, wrong labels).

The AI-training generator allows for a fast and easy-to-implement adaption to identify more and other special handling labels (e.g., 'Do not stack'). With a limited number of real pictures and a vast number of generated images, good results could be achieved. This is especially important as labeling, and the respective handling requirements can change rapidly, as the example of lithium batteries has shown.

Single-piece volume detection without having to manually separate the piece helps to improve transport planning, volume-based billing, and load planning. The integration of the measurement process in the material flow without interruption and reliable near-real-time results saves time and space in the warehouse.

The detection and identification of the 2D package information were proven to be very reliable. The precision of DG label detection could be further increased by examining the detected label areas concerning text existence. For example, an attempt could be made to read the class of the label from the given caption. Furthermore, extending the system with additional cameras could make more regions of the cargo accessible and thus increase the detection rate.

Our approach to automatically generate 3D package annotations was successfully tested with a common point cloud segmentation system (BoNet). Further testing is needed to determine the reliability of the overall system. In particular, scenarios with densely

AI-based recognition of dangerous goods labels and metric package features

packed pallets need to be investigated. Thereby, the required training sets can be generated with little effort using the presented approach.

Acknowledgements

Financial Disclosure

References

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M. and Savarese, S., 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. In: IEEE Conference on Computer Vision and Pattern Recognition. pp.1534–1543.

Borstell, H., 2018. A short survey of image processing in logistics. In: 11th International Doctoral Student Workshop on Logistics. pp.43–46.

Brylka, R., Schwanecke, U. and Bierwirth, B., 2020. Camera Based Barcode Localization and Decoding in Real-World Applications. In: International Conference on Omnilayer Intelligent Systems. pp.1–8.

Creusot, C. and Munawar, A., 2015. Real-Time Barcode Detection in the Wild. In: IEEE Winter Conference on Applications of Computer Vision. pp.239–245.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M., 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition, pp.2432–2443.

David Gilbert and Huy Cuong Nguyen, 2021. ZXing-C++ ("zebra crossing") is an open-source, multi-format 1D/2D barcode image processing library implemented in C++. Available at: <https://github.com/nu-book/zxing-cpp>.

Edlinger, R., Zauner, G. and Zauner, M., 2019. Hazmat label recognition and localization for rescue robots in disaster scenarios. Electronic Imaging, pp.461–463.

Ellis, J., 2010. Undeclared dangerous goods - Risk implications for maritime transport. WMU Journal of Maritime Affairs, pp.5–27.

Fang, J., Zhou, D., Song, X. and Zhang, L., 2021. MapFusion: A General Framework for 3D Object Detection with HDMaps. CoRR, Available at: <https://arxiv.org/abs/2103.05929 >.

Figueiredo, R., Dehban, A., Moreno, P., Bernardino, A., Santos-Victor, J. and Araújo, H., 2019. A robust and efficient framework for fast cylinder detection. Robotics and Autonomous Systems, pp.17–28.

AI-based recognition of dangerous goods labels and metric package features

Gallo, O. and Manduchi, R., 2011. Reading 1D Barcodes with Mobile Phones Using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.1834–1843.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. and Medina-Carnicer, R., 2015. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. Pattern Recognition, pp.481–491.

Goron, L.C., Marton, Z.-C., Lazea, G. and Beetz, M., 2012. Robustly Segmenting Cylindrical and Box-like Objects in Cluttered Scenes using Depth Cameras. In: ROBOTIK German Conference on Robotics. pp.1–6.

Hansen, D.K., Nasrollahi, K., Rasmusen, C.B. and Moeslund, T.B., 2017. Real-Time Barcode Detection and Classification using Deep Learning. In: International Joint Conference on Computational Intelligence. pp.321–327.

Lang, A., Vora, S., Caesar, H., Zhou, L., Yang, J. and Beijbom, O., 2019. PointPillars: Fast Encoders for Object Detection From Point Clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp.12689–12697.

Lienhart, R., Kuranov, A. and Pisarevsky, V., 2003. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In: Proceedings of the 25th Pattern Recognition Symposium. pp.297–304.

Lowe, D., 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, pp.91–110.

Mohamed, M.A., Tünnermann, J. and Mertsching, B., 2018. Seeing Signs of Danger: Attention-Accelerated Hazmat Label Detection. In: IEEE International Symposium on Safety, Security, and Rescue Robotics. pp.1–6.

Namane, A. and Arezki, M., 2017. Fast Real Time 1D Barcode Detection From Webcam Images Using the Bars Detection Method. In: World Congress on Engineering. pp.501–507.

Nations, U., 2019. UN Recommendations on the Transport of Dangerous Goods, Model Regulations. UN Model Regulations. [online] Available at: <https://unece.org/about-recommendations>.

Pham, Q.-H., Nguyen, D., Hua, B.-S., Roig, G. and Yeung, S., 2019. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds With Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp.8819–8828.

Pozo, A.P., Zhao, B., Haddad, A.W., Boutin, M. and Delp, E., 2013. Hazardous material sign detection and recognition. IEEE International Conference on Image Processing, pp.2640–2644.

Preparata, F.P. and Hong, S., 1977. Convex Hulls of Finite Sets of Points in Two and Three Dimensions. Communications of the Association for Computing Machinery, pp.87–93.

Redmon, J., Divvala, S.K., Girshick, R.B. and Farhadi, A., 2015. You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.779–788.

Romero-Ramirez, F., Muñoz-Salinas, R. and Medina-Carnicer, R., 2018. Speeded Up Detection of Squared Fiducial Markers. Image and Vision Computing, pp.38–47.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention, pp.234–241.

Sharifi, A., Zibaei, A. and Rezaei, M., 2020. DeepHAZMAT: Hazardous Materials Sign Detection and Segmentation with Restricted Computational Resources. Machine Learning with Applications, pp.100-104.

Shi, S., Wang, X. and Li, H., 2019. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp.770–779.

Sörös, G. and Flörkemeier, C., 2013. Blur-Resistant Joint 1D and 2D Barcode Localization for Smartphones. In: Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia. pp.1–8.

Tijtgat, N., Volckaert, B. and Turck, F. de, 2017. Real-Time Hazard Symbol Detection and Localization Using UAV Imagery. In: IEEE 86th Vehicular Technology Conference. pp.1–5.

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A. and Trigoni, A., 2019. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. CoRR, Available at: <http://arxiv.org/abs/1906.01140>.

Zamberletti, A., Gallo, I. and Albertini, S., 2013. Robust Angle Invariant 1D Barcode Detection. In: 2nd IAPR Asian Conference on Pattern Recognition. pp.160–164.

Zhang, H., Shi, G., Liu, L., Zhao, M. and Liang, Z., 2018. Detection and identification method of medical label barcode based on deep learning. In: 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA). pp.1–6.

Zhang, Z., 1999. Flexible camera calibration by viewing a plane from unknown orientations. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. pp.666–673.