Using Camera-Drones and Artificial Intelligence to Automate Warehouse Inventory

René Kessler[a], Christian Melching[a], Ralph Goehrs[b] and Jorge Marx Gómez[a]

[a]University of Oldenburg, Ammerländer Heerstr. 114-118, Oldenburg, 26129, Germany
[b]abat AG, An der Reeperbahn 10, Bremen, 28127, Germany

Abstract

Inventory is a very important, but also a very time-consuming manual process in warehouse logistics. This paper presents an approach to automate manual inventory using a camera-drone and various AI procedures. Thereby, sensor technology, such as RFID, is avoided, and only the visual representation of the products and goods is used. We developed a custom dataset that was used for the training of an object detection model to extract and count all relevant objects based on an image of the warehouse. Furthermore, we can show that different pre-processing steps and especially image augmentation methods can significantly influence the performance of such models.

Keywords

inventory, logistics, artificial intelligence, object detection, drones

1. Motivation and problem statement

Logistics plays a crucial role in today's global economy - every company depends on reliable and intact logistics processes to create value. At the same time, logistics is subject to very high margin pressure. On the one hand, improved service is expected from logistics service providers, but on the other hand, customers do not want to to pay extra for it [1]. If the margin is to be maintained or even increased, this can only be achieved through savings in internal processes. For many companies, digital transformation and the use of so-called smart technologies can be the solution for optimizing processes and procedures [2, 3, 4, 5], since many logistics processes still involve a lot of manual effort, including inventory, which is essential for every company [6]. In practice and science, this is often referred to as Industry 4.0. Four main design principles apply to use cases in this area: Networking, information transparency, technical assistance and decentralized decisions [7]. The approach pursued in this work can also be classified in these principles. Through the combined use of camera drones and methods for processing the data, such as AI, human abilities can be imitated: The processing of visual signals. The result is that previously manual activities, such as the inventory process in a warehouse, could be automated.

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.

rene.kessler@uol.de (R. Kessler); christian.melching@uol.de (C. Melching); ralph.goehrs@abat.de (R. Goehrs); jorge.marx.gomez@uol.de (J.M. Gómez)
0000-0002-6426-9266 (R. Kessler)

The goal of the inventory is to record the type and number of internal goods and to quantify this inventory with an exact value [6]. The actual process of inventory can differ not only in company-specific factors but also in the fact that two types of inventories are common. Thus, a distinction can be made between physical inventory, i.e. counting physical goods, and the non-physical inventory, where, for example, financial goods or bank balances are recorded. In this paper, further focus will be on physical inventory. According to the German Commercial Code (HGB), German companies are obliged to carry out an inventory atleast once a year[1] (paragraph 240 German Commercial Code). While this cycle may be sufficient for companies with little movement of goods, it makes sense to keep shorter cycles especially for companies in the retail sector. However, the inventory can also be understood as an instrument of quality management concerning transparency in order to be able to monitor the business goals and their achievement. With the help of an inventory, deficits, faulty processes, non-optimal flows, or even theft can be detected. During an inventory, there is always a large amount of personnel effort involved. Depending on the inventory's size and scope, several people are often exclusively occupied with the manual counting of the goods. Inventory not only causes personnel costs, but also disrupts operational processes. Therefore, companies find themselves in a dichotomy between transparency and costs, which is why they often work with samples whose findings can then be extrapolated to the entire inventory. This business conflict can be resolved by automating the manual localization and identification of products and goods during inventory, resulting in greater transparency at lower costs [8]. Existing approaches are based on the use of sensor technology and often only consider a very specific sub-area (e.g. industries). In this paper, a generic approach is followed, which exclusively involves the visual representation of the products and is based on deep learning methods. To also automate the acquisition of the images and thus to evaluate a vehicle for the operationalization of the approach, a drone is also used, which has already proven itself in comparable applications [9, 4, 10, 11]. Combined with the current problems in inventory processes described above and the great potential through automation of these process steps, this leads to the research question:

Which AI procedures are suitable to recognize products on images in order to count them for an inventory?

To solve the described problem, a data-driven approach was followed, which was oriented towards the established CRISP-DM model [12, 13]. During implementation, the company cooperated intensively with a North German beverage distributor, which has several storage locations and a high volume of trade in the B2B sector. Initially, domain knowledge about the storage locations and the inventory process was collected in several workshops and discussions. Also, we were granted access to the different warehouses in order to record the custom dataset (3). Finally, the results of the experiments were presented to the practice partner, and practical implications were derived.

The paper is structured as follows: In the next section, *Related Work* (2), an overview of the current research work is given. Section three, *Dataset* (3), is dedicated to the structure of the specially compiled dataset and its annotations. The fourth section, *Experiment and Results*

[1] https://www.gesetze-im-internet.de/hgb/__240.html, accessed on 20.11.2020

(4), represents the core of the work and describes the methods used and their results. In the concluding section, *Discussion and Future Work* (5), the results are summarized, and an outlook on further work is given.

2. Related work

A search for related work has led to identifying numerous publications that deal with drones in logistics and use of digital technologies to optimize or even automate the inventory process. Two main lines of research have been identified and are presented below.

Locating and identification utilizing sensor technology: Radio-Frequency Identification (RFID) tags are a widespread and established method to track products and load carriers in a warehouse [14, 15, 16], and to achieve an increase in transparency regarding warehouse movements [17]. Often RFID readers are attached to a drone, and the drone flies over storage locations and tracks the individual goods [18, 15, 19, 9, 3, 20, 8]. Drones can detect storage locations that would otherwise be difficult to reach (e.g., very high storage locations in a high rack) and minimize the risk for the employees [16, 21]. In summary, the identification of loads using RFID, especially in combination with a drone, has proven to save costs and automate inventory processes [17]. However, this always presupposes that the loads are equipped with RFID tags, representing a further cost factor in logistics.

Reading optical product features or characteristics: There is a necessary consensus that image processing in logistics can be a great advantage for the traceability and monitoring of goods [22]. If camera-drones are used in inventory management, they are often used as a medium for reading optical product annotations, such as one-dimensional barcodes or QR-Codes [23, 24, 8]. However, these approaches assume that those annotations exist. Considering packed products, barcodes are often available and do not pose a problem. But with unpackaged goods or empties, optical annotations are rarely present. Here is a need for solutions that focus on the optical representation of goods. Although both AI-based image processing and the use of drones are considered to have great potential in logistics [25, 10, 8], there are hardly any publications available that combine these two approaches. Freistetter and Hummel (2019) outlined an approach to drone-based inventory in libraries. They flew off bookshelves and identified book spines using computer vision techniques. As soon as a book is in the center of the image, the title of the book is read [26]. This is a particular indoor use case. Despite the fundamental similarity, a use case from a library cannot necessarily be transferred to larger industrial warehouses. Especially concerning disruptive factors (e.g. environmental influences, such as changing weather, which can lead to different lighting), recordings in libraries are less affected. Dörr et al. show an approach that deals with a similar use case in the warehouse area. The goal of the approach is product structure recognition based on image data. Different convolutional neural networks are used based on top of each other. For the training of the models a separate dataset was built [27]. The very recent publications show that the combination of drone-based inventory and AI image processing is currently subject of research [16].

Especially the approach of Dörr et al. shows many similarities to our approach presented

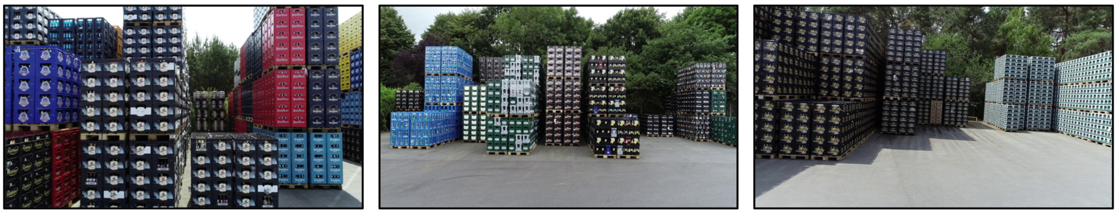


Figure 1: Exemplary sample from the final dataset

here, e.g., the hierarchical structure with several deep learning models [27]. However, there are also several differences. The approaches differ in their place of use since this thesis is an outdoor use case. Also, the type of objects to be identified differs.

3. Dataset

The use case considered here focuses on the automated recognition of beverage pallets on images. Since there is no public dataset for this specific problem, we developed a custom dataset. The structure and further processing of the dataset is described in the following.

Data Acquisition: A drone of the model *DJI Phantom pro 4 v2*[2] was used to make the video recordings in the warehouses of the practice partner. The drone was selected because it was already available for the research project, so there was no need to purchase a new drone. In addition, the characteristics of this drone are very common, which is why the findings are equally transferable to other drone models. The video recordings were made in Full-HD resolution, a frame rate of 30 frames per second and using the built-in image stabilization. To create the greatest possible variance in the data, the recordings were made over four days and at two locations of the beverage dealer. The aim was to record the weather's influence on the images (e.g., brightness) by recording the images under different conditions. During the project, two days with sunny weather and one day each with cloudy and very cloudy weather were used for the recording. Special attention was also paid to the recorded scenes. We tried to capture all pallet locations of the outdoor warehouse. It could not be avoided that not all types of pallets (e.g., different manufacturers of beverages) are represented equally often in the data because the different pallets' stock varies very much in reality.

After finishing the video recordings, the video data had to be processed. For this purpose, the filmed sequences were viewed manually, and irrelevant scenes were removed (e.g., the drone's starting and landing sequences). Since there are only marginal differences in subsequent frames at 30 frames per second, only every 30th frame of the video clips was transferred to the final data set when the training data set was created to avoid potential overfitting when training the neural network. After the frames' automated extraction, they were manually sighted, and faulty or low-quality images were removed. The final dataset consists of 336 images, separated into train and test set. The test set only contains images from pallet stacks that are not included in the train set to avoid memorization. This includes pallets with beverages of previously un-

[2] https://www.dji.com/de/phantom-4-pro-v2/specs

seen brands and breweries.
Data Annotation: After image acquisition, annotation has been applied using the tool *label-studio* from *Heartex*[3]. This tool was used because it offers many possibilities for annotating images and was already used in another context within the research project. In this process, each pallet that was largely visible was annotated using polygons instead of only bounding boxes to make use of the annotation masks later on. Additionally, each polygon was given a class to differentiate between two types of pallets, pallets containing cases of beer and pallets containing other beverages. The resulting annotations were exported and converted to the COCO dataset format [28], as it is one of the standard formats for object detection and segmentation in images that are widely supported by most frameworks. This results in a training set containing 284 images with 5261 annotated polygons and a test set containing 52 images and 1471 polygons.

4. Experiment and Results

The experiment was conducted as follows. First a baseline model was used to test the impact of various modifications of the input data on the prediction accuracy. Then, several models using different architectures, selected based on defined criteria, were trained to evaluate their performance applying the identified modifications using the baseline model. Finally, the best performing model was used to perform a qualitative evaluation and to identify possible errors.

4.1. Baseline Model

The model used during the following experiments is a Mask R-CNN using a ResNet50 Backbone, implemented using Detectron2 [29, 30]. The model was pre-trained using the MSCOCO17 dataset to try to make up for the low amount of images in the used dataset, described in 3. It was then trained over up to 4000 iterations, where each iteration used a batch of twelve images. Evaluations of the model were performed every 250 iterations during training, and once the training was completed. During training and testing, the images were resized to 1000x750 and not further modified. To validate each experiment's results, it was repeated multiple times; the following metrics are averages over all runs. For all experiments the same train-test-split was used.

4.2. Initial Results and Adjustments

The initial baseline model achieved an average precision of 27.06 and 27.59 evaluating bounding boxes and segmentation respectively, using the test portion of the dataset. As these results are far from sufficient to predict the pallets' position on images accurately, significant adjustments were necessary to improve the performance of the model.

[3] https://labelstud.io

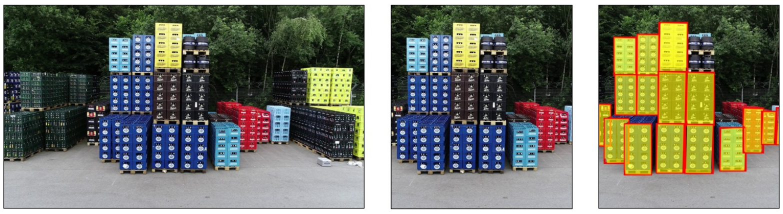


Figure 2: Example of an image (left), its simplified version (middle) and its annotations (right).

4.2.1. Merging of classes

During the first tests, it became clear that the model had difficulties in classifying the pallets given the annotated classes (beer and other beverages). Not only was the per-class-precision much higher on the pallets marked as containing cases of beer (33.647 compared to 20.463, evaluating bounding boxes), some pallets were often wrongly classified. While this problem's origin likely lies within the training data that contained more annotations and variants of pallets of beer cases, it is challenging to balance the classes to reduce this spread since nearly all images contain pallets of both categories. The classes have been merged to create a model to predict only the bounding box and segmentation mask of a pallet, not further classifying its contents to circumvent this problem. If applied like this, the classification task must be processed by a different system, possibly using classifier or brand detectors. This work has not yet further pursued the creation of such a solution.

A model trained on a classless dataset achieves an average precision of 45.95 and 46.70 on bounding boxes and masks, as noted in table 1. A following manual inspection of the predictions also confirmed the increase of the quantity and quality of the predictions.

4.2.2. Reduction of image area

Another problem of the model was the detection of small pallets on the edges of the images. It is likely caused by the low amount of small objects in the training data and distortion of the camera lens on the edges of the image. This problem can be addressed by cutting off parts on the left and right edges of the recordings. This should not harm the quality of the inventory process as the drone flight is planned to fly so that each row is at least once near the center of the recorded images and therefore not lost in this process. An example of such reduction of the image contents is visible in figure 2 where 50% of the image have been removed, in equal parts on each side. During the training and evaluation process, the input images were resized to 750x750 instead of 1000x750, to better match the aspect ratio of the modified images. The resulting model achieves values far better than before the removal. This is expected as the problem is simplified significantly. The model achieves a mean Average Precision (mAP) of 47.68 and 46.70 evaluating predicted bounding boxes and segmentation, respectively.

4.2.3. Image augmentation

Image augmentation is widely used in many research projects [31, 32, 33], but can have different effects depending on the problem [27]. Therefore, different image augmentation methods were tested and evaluated based on the performance of the models trained using augmented images. The augmentation of the original images was applied using *imgaug*[4] during the loading of the image batch of each iteration. Each image of the batch was augmented individually with random parameters in given boundaries. As *imgaug* offers a wide variety of methods to augment images, some were selected to evaluate its performance on the given problem. In this case we chose augmentations to simulate variations of real recordings, such as *MotionBlur*, *PerspectiveTransformation*, *Contrast* and *JpegCompression*, to accommodate for movement of the drone, varying lighting and weather conditions, while also considering methods used traditionally to adapt the image, *Rotation*, *ScaleXY*, *FlipLR* and *CropAndPad*. The following methods were chosen for comparison.

(a) **FlipLR** - Performs a horizontal flip with a probability of 0.5.
(b) **ScaleXY**{150,125} - Scales the width and the height of the images with using random values from [0.5, 1.5] for ScaleXY150 and [0.75, 1.25] for ScaleXY125.
(c) **Rotate**{10,20,30} - Rotates the image around its center using random degrees up to 10 for Rotate10, 20 for Rotate20 and 30 for Rotate30. Rotations can applied in both directions.
(d) **Contrast** - Increases or decreases the contrast of the image using random values from [0.5, 1.4].
(e) **JpegCompression** - Reduce the quality of the image by applying Jpeg compression using a random degree from [0.7, 0.95].
(f) **MotionBlur** - Creates a motion blur effect with a kernel of size 7x7.
(g) **CropPad**{25,50} - Remove random percentage of images from all edges and pad the image to its original size. Using random values from [-0.25, 0.25] for CropAndPad25 and from [-0.5, 0.5] for CropAndPad50.
(h) **PerspectiveTransform** - Transforms the image, as if the camera had a different perspective using random scales from [0.01, 0.1].
(i) **FlipLR-ScaleXY150** - Combines FlipLR and ScaleXY150.
(j) **CropPad25-ScaleXY150** - Combines CropAndPad25 and ScaleXY150.
(k) **Rotate20-ScaleXY150** - Combines Rotate20 and ScaleXY150.

An overview of the effect of these augmentations is displayed in figure 3. Some of the tested augmentations were omitted in the figure as their effects are barely visible due to the size of the individual images or are variations or combinations of already displayed effects.

It has to be noted that after the application of one or multiple augmentations the bounding boxes were recomputed according to the transformed mask, to make them a minimal fit to the object again. Otherwise, some augmentations, such as Rotation, could create a bounding box according to the transformed bounding box, which could be too large to accurately locate the object. In some situations, this results in a small, but measurable, boost of performance of models using affected augmentations.

4.2.4. Baseline results

To evaluate the different possible modifications, we trained several models using the same settings and evaluated them on the shared test set. For the evaluation of the different methods,

[4] https://github.com/aleju/imgaug

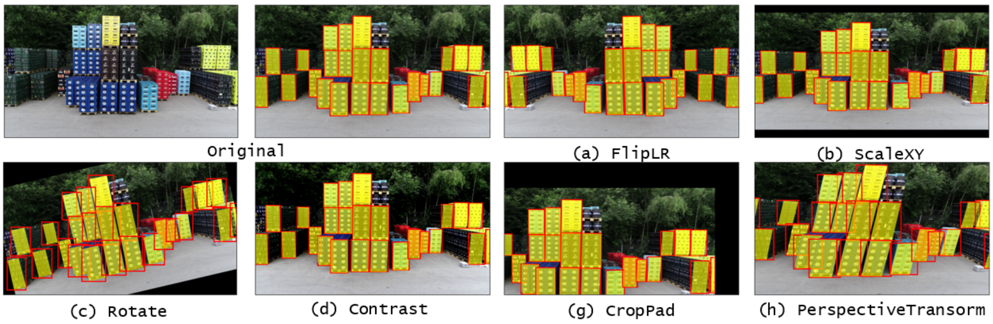


Figure 3: Examples of results using various selected augmentation methods, applied to "Orignal".

we compared both the mAP of the predicted bounding boxes and the mAP of the segmentations, even though they are very similar, displayed in table 1.

The best performing model without the use of image augmentation was the model trained using the simplest variation of the images, utilizing the cutting of edges and merging of different classes. It achieved a precision of more than 70 and therefore performs better than most other models, including many models trained using image augmentation. Once image augmentation (Table 1) is considered, the model trained on unaugmented images is outperformed by several different models. While nearly all models using merged classes and cut images perform better, only a few models using a different image base produce comparable or better results.

Especially interesting are the effects of specific augmentation methods. While the Rotation augmentation decreased the accuracy of models using images with uncut edges, it increased the precision on images scaled to a 1:1 aspect ratio. Some methods seem almost always to reduce the prediction quality, such as Contrast (d), JpegCompression (e), and MotionBlur (f). While the idea behind using these methods was to make images slightly more corrupt to increase the ability to learn from realistic variations of these images, it mostly hurt performance. Other augmentation methods used, such as ScaleXY (b), FlipLR (a), and CropPad (g), seem to always improve the results of the trained model when used alone or in combination with other methods, contrary to the observation by Dörr et al.. This is supported by the fact that the almost always best-performing method used the combination of ScaleXY (b) and FlipLR (a).

4.3. Evaluation of other architectures

While the results provided by the different Mask R-CNN models certainly provide valuable information, the model itself is no longer state-of-the-art in terms of precision. Therefore we selected three different models and tested their performance using the results gained using the previous model. The first model we additionally tested is DetectoRS [34], whose innovative characteristic is the use of Recursive Feature Pyramids. It achieves near state-of-the-art performance on the MSCOCO17 dataset and was implemented and trained using MMDetecion [35]. The second model selected is Yolact [36]. Yolact is an architecture that is able to generate predictions of the recordings in near real time and was trained using the MMDetection framework aswell. While real time predictions are not necessary during the stocktaking

Table 1
Results of various experiments and combinations of possible modifications. Each entry “A/B” denotes the mAP of the bounding boxes and the segmentation respectively.

		Original	Class Merging	LeftRight-Cut	Merging+Cut
	No augmentation	27.06/27.59	45.95/46.70	47.68/48.93	70.39/72.19
(a)	FlipLR	30.03/31.13	46.58/47.48	53.58/55.44	73.94/76.01
(b)	ScaleXY125	45.01/46.27	67.57/69.28	54.62/56.45	78.97/81.06
	ScaleXY150	49.83/51.00	73.35/75.54	56.01/57.49	78.86/80.93
(c)	Rotate10	25.48/26.23	41.89/42.81	52.87/53.46	77.07/77.63
	Rotate20	25.31/26.06	42.96/43.95	50.11/51.35	76.08/76.71
	Rotate30	24.95/25.83	43.56/44.99	47.49/48.43	75.57/76.04
(d)	Contrast	26.35/26.89	40.22/40.71	48.97/50.05	68.69/70.15
(e)	JpegCompression	26.12/27.06	42.21/43.29	49.35/50.65	67.83/69.35
(f)	MotionBlur	25.27/26.30	43.61/44.76	44.89/46.30	71.19/73.01
(g)	CropPad25	47.22/48.51	70.81/72.89	55.63/57.39	78.68/80.58
	CropPad50	46.80/47.91	68.97/70.92	52.47/54.07	75.50/77.28
(h)	PerspectiveTransform	31.54/32.10	53.13/53.99	54.10/55.42	78.11/79.30
(i)	FlipLR-ScaleXY150	**51.33/52.71**	**73.84/75.78**	57.09/58.85	**81.53/83.41**
(j)	CropPad25-ScaleXY150	47.08/48.30	71.25/73.24	**57.99/59.28**	78.13/79.67
(k)	Rotate20-ScaleXY150	46.71/47.76	70.92/72.57	56.12/56.50	79.51/79.93

Table 2
Performance values of different models using Merging+Cut and augmentation (i).

Model	mAP (bbox/segm)	Inference time (GPU)	Inference time (CPU)	Parameters
Mask R-CNN	81.5/83.4	**0.04 s**	1,72 s	43,937,313
DetectoRS	**88.3/85.6**	0.13 s	Not supported	131,648,615
Yolact	50.3/60.6	0.05 s	**0.59** s	**34,727,123**
DETR	74.8/81.8	0.12 s	2.10 s	42,835,552

process, Yolact was selected since these models could easily be used to serve different purposes within the same domain. The third and last model evaluated is DETR [37] due to it’s innovative approach. DETR utilizes the transformer architecture introduced in the domain of NLP to generate instance segmentations. It was chosen to evaluate whether or not new and innovative approaches can be applied to the domain of palettes, and trained using the code provided by the authors[5].

4.3.1. Results

Each of the additional models was tested and evaluated on the test dataset, using the merged and cut variant and the augmentation method (i) that showed to increase performance the most. The results are displayed in table 2. It is clear that DetectoRS outperforms all other models by a significant margin in terms of precision. It achieves a mAP of 88.3 and 85.6 while taking 0.13 s per frame using a NVIDIA® GeForce® RTX 2080 Ti, making it also slower than all other models. In contrast, Yolact achieves the lowest mAP and, contrary to expectations,

[5] https://github.com/facebookresearch/detr

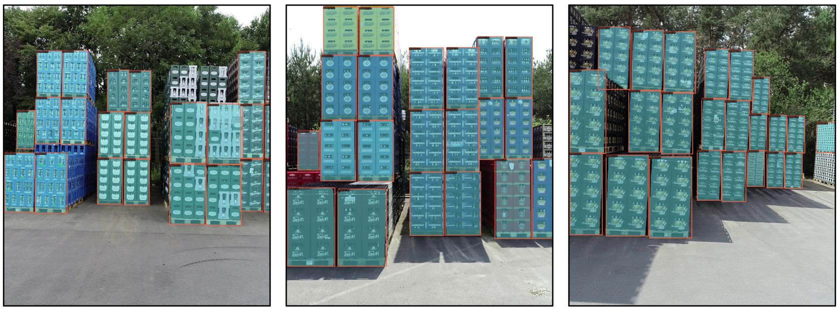


Figure 4: Exemplary predictions using the best performing model, DetectoRS, according to table 2

is not the fastest model, but is 0.01 s slower than Mask R-CNN. However, Yolact is by far the fastest model when using a CPU. DETR, with it's new approach, is both both slower and less precise than Mask R-CNN and does not stand out in any metric.

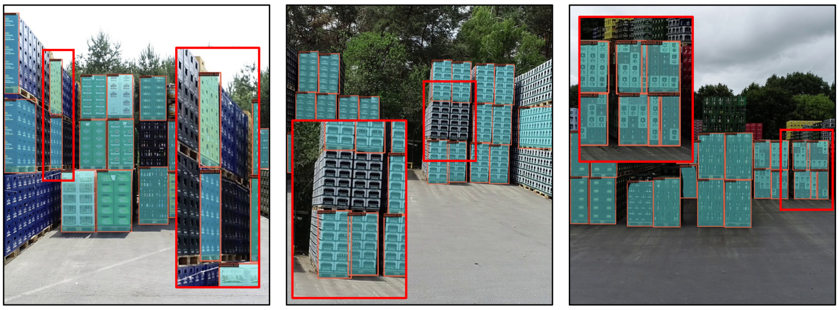
In addition to evaluation based on metrics, timings, and model size, a manual qualitative evaluation was performed. This showed that the mAP value was consistent with the visual impression. In terms of both bounding boxes and segmentation, DetectoRS provides the best results. Mask R-CNN also delivers satisfactory results, while the quality of DETR and Yolact in particular falls off sharply.

To get an impression of the quality of the predictions of DetectoRS, some of its predictions are visualized in figure 4. The generated predictions are predominantly of high to very high quality. In a few cases, however, there are (partly) incorrect predictions (figure 5). Three typical errors can be described as follows:

1. **Recognition of side views of pallets:** Despite the label strategy and the pre-processing steps, in the case of images with a very specific acquisition angle, namely whenever the side views occupy a large image area, the isolated, incorrect identification of pallets occurs, in which side views are provided with a bounding box and mask.
2. **Individual pallets are not recognized:** The evaluation has shown that in rare cases individual pallets are not recognized. The special feature here is that this error always refers only to a maximum of two pallets standing next to each other. All other pallets on these images were recognized completely and without errors. Further optimization of the detector parameters (e.g., tresholding) will most likely solve this error.
3. **Strongly overlapping bounding boxes:** Pallets with boxes of different colors sometimes have overlapping bounding boxes. This has no consequences for the pallet recognition, but it could lead to problems in subsequent steps, such as the classification of the pallets. To solve this problem, further methods could be used in the preprocessing of the images.

5. Discussion and Future Work

In this work, we present an initial step for automatized inventory using images recorded by a drone and an AI based object detector to identify the location of pallets on recorded images.



Recognition of side views of pallets

Individual pallets are not recognized

Overlapping bounding boxes

Figure 5: Visualization of Predictions with Errors

Various modifications have been tested to increase the accuracy of predicted bounding boxes and segmentation masks, partly without compromising the quality and direct usability of the results. In doing so, we also showed that image augmentation methods could increase the precision of models significantly, contrary to the observations in related projects [27]. In summary, it has been shown that horizontal flipping and image scaling as an image augmentation technique can have a positive impact on performance during model training. In particular, the model architecture DetectoRS showed very good results in the experiments, measured by mAP (bounding boxes and segmentation), although not being the fastest in comparison. While already delivering promising results, there are certain factors limiting the usage of the developed models in practice.

Firstly, a solution for the classification of the type or class of beverages on the pallets must be developed and integrated. While it would have been preferable to predict its class using the same model that also predicts its location, tests showed it impacted its localization performance significantly. Secondly, while the detection of the front of the pallets gives valuable information to automate the inventory process, additional information is needed to complete it. This mainly includes the length-wise number of stacks of pallets in their row, which could be recorded from above. Finally, regardless of the technical implementation, discussions and tests with practical partners have shown that the organization in the warehouse must also be changed if drones are to be used. Even though drones are very flexible and can reach storage areas that are difficult for humans to reach, processes in the warehouse must be adapted to ensure that drones can be used safely and efficiently [38].

References

[1] M. Hompel, B. Otto, Essay zur Logistik 4.0 (2015). doi:10.13140/RG.2.1.2857.4245.

[2] pwc, Five forces transforming transport logistics, 2019. URL: https://www.pwc.pl/pl/pdf/publikacje/2018/transport-logistics-trendbook-2019-en.pdf.

[3] C. Cimini, A. Lagorio, F. Pirola, R. Pinto, Exploring human factors in logistics 4.0: empirical evidence from a case study, IFAC-PapersOnLine 52 (2019) 2183 – 2188. URL: http://www.sciencedirect.com/science/article/pii/S2405896319315137. doi:https:

//doi.org/10.1016/j.ifacol.2019.11.529, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

[4] M. Maslarić, S. Nikolicic, D. Mirčetić, Logistics response to the industry 4.0: The physical internet, Open Engineering 6 (2016). doi:10.1515/eng-2016-0073.

[5] E. Companik, M. J. Gravier, M. T. F. II, Feasibility of warehouse drone adoption and implementation, Journal of Transportation Management 28(2) (2018) 33–50.

[6] H. Gleissner, J. C. Femerling, IT in Logistics, Springer International Publishing, Cham, 2013, pp. 189–223. URL: https://doi.org/10.1007/978-3-319-01769-3_9. doi:10.1007/978-3-319-01769-3_9.

[7] M. Hermann, T. Pentek, B. Otto, Design principles for industrie 4.0 scenarios, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), 2016, pp. 3928–3937. doi:10.1109/HICSS.2016.488.

[8] T. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, Fraga-Lamas, Towards an autonomous industry 4.0 warehouse: A uav and blockchain-based system for inventory and traceability applications in big data-driven supply chain management, Sensors 2019 19 (2019) 2394.

[9] C. S. Tang, L. P. Veelenturf, The strategic role of logistics in the industry 4.0 era, Transportation Research Part E: Logistics and Transportation Review 129 (2019) 1 – 11. URL: http://www.sciencedirect.com/science/article/pii/S1366554519306349. doi:https://doi.org/10.1016/j.tre.2019.06.004.

[10] DHL Trend Research, The logistics trend radar (5th edition), 2020. URL: https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/glo-core-logistics-trend-radar-5thedition.pdf.

[11] A. Lotz, Drones in logistics: A feasible future or a waste of effort (2015). URL: https://scholarworks.bgsu.edu/honorsprojects/204.

[12] V. Grover, K. Lyytinen, New state of play in information systems research: The push to the edges, MIS Q. 39 (2015) 271–296.

[13] C. Shearer, The crisp-dm model: The new blueprint for data mining, Journal of Data Warehousing 5 (2000) 13–22.

[14] E. Ilie-Zudor, Z. Kemény, F. van Blommestein, L. Monostori, A. van der Meulen, A survey of applications and requirements of unique identification systems and rfid techniques, Computers in Industry 62 (2011) 227 – 252. URL: http://www.sciencedirect.com/science/article/pii/S0166361510001521. doi:https://doi.org/10.1016/j.compind.2010.10.004.

[15] P. Thanapal, J. Prabhu, M. Jakhar, A survey on barcode rfid and nfc, IOP Conference Series: Materials Science and Engineering 263 (2017) 042049. doi:10.1088/1757-899X/263/4/042049.

[16] D. Cristiani, F. Bottonelli, A. Trotta, M. Di Felice, Inventory management through mini-drones: Architecture and proof-of-concept implementation, in: 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020, pp. 317–322. doi:10.1109/WoWMoM49955.2020.00060.

[17] Y. Y. Cheung, K. L. Choy, C. W. Lau, Y. K. Leung, The impact of rfid technology on the formulation of logistics strategy, in: PICMET '08 - 2008 Portland International Conference on Management of Engineering Technology, 2008, pp. 1673–1680. doi:10.1109/PICMET.

2008.4599787.

[18] P. Jhunjhunwala, M. Shriya, E. Rufus, Development of hardware based inventory management system using uav and rfid, in: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1–5. doi:10.1109/ViTECoN.2019.8899488.

[19] B. Rahmadya, R. Sun, S. Takeda, K. Kagoshima, M. Umehira, A framework to determine secure distances for either drones or robots based inventory management systems, IEEE Access 8 (2020) 170153–170161. doi:10.1109/ACCESS.2020.3024963.

[20] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, S. Behnke, Fast autonomous flight in warehouses for inventory applications, IEEE Robotics and Automation Letters 3 (2018) 3121–3128. doi:10.1109/LRA.2018.2849833.

[21] J. P. Škrinjar, P. Škorput, M. Furdić, Application of unmanned aerial vehicles in logistic processes, in: I. Karabegović (Ed.), New Technologies, Development and Application, Springer International Publishing, Cham, 2019, pp. 359–366.

[22] H. Borstell, A short survey of image processing in logistics - how image processing contributes to efficiency of logistics processes through intelligence, 2018. doi:10.13140/RG.2.2.11060.76168.

[23] I. Kalinov, A. Petrovsky, V. Ilin, E. Pristanskiy, M. Kurenkov, V. Ramzhaev, I. Idrisov, D. Tsetserukou, Warevision: Cnn barcode detection-based uav trajectory optimization for autonomous warehouse stocktaking, IEEE Robotics and Automation Letters 5 (2020) 6647–6653. doi:10.1109/LRA.2020.3010733.

[24] S. Hong-ying, The application of barcode technology in logistics and warehouse management, in: 2009 First International Workshop on Education Technology and Computer Science, volume 3, 2009, pp. 732–735. doi:10.1109/ETCS.2009.698.

[25] L. Wawrla, O. Maghazei, T. Netland, Application of drones in warehouse operations (2019). URL: www.pom.ethz.ch, whitepaper from ETH Zurich (Chair of Production and Operations Management.

[26] A. Freistetter, K. A. Hummel, Human-drone teaming: Use case bookshelf inventory, in: Proceedings of the 9th International Conference on the Internet of Things, IoT 2019, Association for Computing Machinery, New York, NY, USA, 2019. URL: https://doi.org/10.1145/3365871.3365913. doi:10.1145/3365871.3365913.

[27] L. Dörr, F. Brandt, M. Pouls, A. Naumann, Fully-automated packaging structure recognition in logistics environments, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), volume 1, 2020, pp. 526–533. doi:10.1109/ETFA46521.2020.9212152.

[28] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, CoRR abs/1405.0312 (2014). URL: http://arxiv.org/abs/1405.0312. arXiv:1405.0312.

[29] K. He, G. Gkioxari, P. Dollár, R. B. Girshick, Mask R-CNN, CoRR abs/1703.06870 (2017). URL: http://arxiv.org/abs/1703.06870. arXiv:1703.06870.

[30] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2, 2019.

[31] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, CoRR abs/1911.09070 (2019). URL: http://arxiv.org/abs/1911.09070. arXiv:1911.09070.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, A. C. Berg, SSD: single shot multibox detector, CoRR abs/1512.02325 (2015). URL: http://arxiv.org/abs/1512.02325. arXiv:1512.02325.

[33] X. Chen, R. Girshick, K. He, P. Dollár, Tensormask: A foundation for dense object segmentation, 2019. arXiv:1903.12174.

[34] S. Qiao, L.-C. Chen, A. Yuille, Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution, 2020. arXiv:2006.02334.

[35] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, D. Lin, MMDetection: Open mmlab detection toolbox and benchmark, arXiv preprint arXiv:1906.07155 (2019).

[36] D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, Yolact: Real-time instance segmentation, in: ICCV, 2019.

[37] M. Zheng, P. Gao, X. Wang, H. Li, H. Dong, End-to-end object detection with adaptive clustering transformer, 2020. arXiv:2011.09315.

[38] E. H. C. Harik, F. Guérin, F. Guinand, J. Brethé, H. Pelvillain, Towards an autonomous warehouse inventory scheme, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1–8. doi:10.1109/SSCI.2016.7850056.