Feasibility Study - SmartWarehouse

Felix Hausberger

Abstract— **In this short paper the object detectors** *You Only Look Once* **and** *Single Shot MultiBox Detector* **are compared for precision, reactivity, and inference behaviour and examined for their potential for industrial use. The background scenario of** *SmartWarehouse* **offers live video data of a drone with goods in a warehouse, which are to be classified and localized in real time. In the future, this should make it possible to carry out inventories and inventory analyses of a warehouse in a time- and cost-efficient manner conserving resources. The goal of this feasibility study is also to find out whether the** *SmartWarehouse* **scenario is technically feasible.**

I. INTRODUCTION

In times of the industrial change towards Industry 4.0 object detectors represent an optimization potential not to be neglected, e.g. in warehousing and logistics. Combined with an autonomous drone, such object detectors could make it possible to conduct inventory checks in a warehouse without human assistance. How different object detectors behave when applied to a real time industry scenario like *SmartWarehouse* should be evaluated in this short paper. As being a feasibility study, the main goal of this work also is to discuss the technical feasibility of the *SmartWarehouse* idea. In section II related work to the two main object detectors should be introduced before explaining the approach and architecture of the *SmartWarehouse* prototype in section III. Next, some key aspects during the realization will be considered in section IV. In section V the results of the feasibility study will be presented and evaluated before the interpretability of the results is discussed in section VI. section VII gives a quick conclusion about the main achievements of this short paper.

II. RELATED WORK

The idea of automized inventory checks with drones is not new. [1] uses a similar approach, but uses normal RFID technology or simple barcodes to identify a product. Using this approach only a small number of instances at a time can be identified, while using object detection algorithms enable many-numbered and faster processing.

The paper [2] introduces the *Single Shot MultiBox Detector* (SSD), an object detector reaching up to 76.9% mean average precision (mAP) while still keeping real time detection characteristics compared to its preceeding competitors like regional convolutional neural networks (R-CNNs). It uses a single neural network to detect objects and creates a set of default bounding boxes over different aspect ratios and scales for each feature map location instead of using an additional bounding box proposal step.

The *You only Look Once* (YOLO) algorithm was introduced in [3] and as well uses a single neural network to predict bounding boxes and class probabilities. Incremental improvements led to version three of YOLO, which is equally accurate as SSD, but three times faster [4].

III. ARCHITECTURE OF THE SMARTWAREHOUSE SCENARIO

A. Defining the scenario

Two goals were defined for the feasibility study of *SmartWarehouse*. First it is to be evaluated, how well existing object detectors perform in industry scenarios taking the example of *SmartWarehouse* scenario. Second, one should deal with the feasibility of the implementation of this *SmartWarehouse* scenario, which comprises the execution of an inventory for department stores with a drone.

B. Dataset

In the construction of the training dataset, the feasibility study does not aim to completely represent a large department store in the dataset. To prove the general feasibility of the *SmartWarehouse* scenario, it is extensive enough to restrict the focus on unpacked beverage bottles of a department store which can be assigned to nine different classes. All classes are distributed equally among the 1088 manually annotated images. 75% of the images include a single instances of a beverage bottle to better train the patterns of the different classes. Furthermore, 13% of the images contain occlusions of objects to be detected, detection situations from extreme viewing positions are represented to 4%, and difficult illumination and lighting conditions are included to 8%. Only a few examples (2%) are related to different distances during detection. Care was also taken to continuously vary the backgrounds of the objects being detected. Finally, to simulate the department store where several objects are to be detected at once, in 12.5% of all images the objects are arranged on shelves, one behind the other or in beverage crates.

C. Introduction of evaluation criteria

To better compare object detectors with each other the following evaluation criteria have been defined:

- Precision: The precision of an object detector measured in mAP
- Reactivity: The inference rate with the model measured in *Frames Per Second* (FPS)
- Inference behavior: The inference behavior under special lighting conditions, extreme viewing positions, occultation of objects, and the behavior with double detected objects measured by the confidence score.

D. Selection of object detectors

For the *SmartWarehouse* scenario, a choice should be made between the four detectors Faster R-CNN, Mask R-CNN, SSD and YOLO.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Fig. 1. Comparison of SSD on PascalVOC 2007 [2]

According to the reference results in Figure 1, it is clear that the SSD performs best in terms of mAP with 74.3% and 76.8%, respectively. Faster-RCNN can keep up with 73.2% in terms of mAP, but with only 7 FPS it is not designed for fast inference. YOLO scores worse than the SSD in both categories, it essentially achieves an mAP of 66.4% and a frame rate of 21 FPS. The SSD thus manages to maintain a good balance between precision and responsiveness. Comparing Mask-RCNN from the original paper in [5] with Faster-RCNN (RoI-Align) results in a mAP difference of 38.2% to 37.3%. Only 5 FPS could be reached.

As the effort to setup and to adapt Faster-RCNN and Mask-RCNN to be trained on a custom dataset was far higher and the performance in time-critical model inference was poorer, YOLO and SSD were selected as the two detectors to evaluate the capability of object detectors for industrial use.

E. Drone selection

Analyzing the market for programmable drones with a freely available software development kit (SDK) and integrated camera, the supply is very low. The choice restricts to either the *Parot Bebop II* or the *Ryze Tello EDU*. When also taking into account the legal framework, only the *Ryze Tello EDU* was left for legal usage.

F. Inventory software specification

To process the drone's image data for an inventory, the drone is to be accessed by a client application. The server, acting as a client, is to connect to the drone and sends the flight signals as static flight instructions to the drone. The deep learning model is also to be integrated on the server for inference. The server's task will be to infer the video stream frames received from the drone with the model, count the detected objects for the inventory using a counting algorithm, and then forward the image data with the bounding boxes drawn in via livestream to a web application for visualization. A REST interface should provide information about the inventory data of the warehouse and provide a way to initiate the flight sequence.

IV. REALIZATION

SSD. The base network of the SSD consists of a VGG16 pre-trained on ImageNet. The remaining convolutional layers are Xavier initialized. SSD was trainied with a batch size of 16, $1.0e^{-3}$ as learning rate and 0.9 as momentum. ReLU was chosen as its activation function and smooth L1 as its loss function. Using the mini-batch learning approach training was done with six-fold cross validation with 22 epochs each.

YOLO. YOLO was trained with a batch size of 64 and 16 subdivisions. The learning rate was $1.0e^{-3}$ initially, then decreased to $1.0e^{-4}$ after 80%, and to $1.0e^{-5}$ after 90% of the training process, the momentum was also 0.9. 18.000 batches were specified in total for the mini-batch gradient descent training process. LReLU was chosen as the activation function and squared errors as the loss function.

Counting algorithm. After each inference process, the detected objects are saved so that the algorithm can access both the most recently detected objects and the newly detected objects when it is called. For all detected objects in the current run, the algorithm searches all last detected objects to find out if an object was already present in the previous image. Crucial for this is the same label as well as the distance of the bounding boxes of the currently detected object to the object of the previous round. Only if the object was not already present in the previous round according to these criteria, it is counted.

Drone connection. For control and access to the video stream, the Ryze Tello EDU offers its own WiFi network to which the control unit must connect. The communication takes place via the UDP protocol and consists of several communication channels. The sending of flight commands and the polling of the video stream is done via a bidirectional socket connection between the client application and the Ryze Tello EDU.

Model inference. Since the H.264 decoder can only run on Python version 2.7 and the PyTorch framework used for the SSD or the Darknet framework for YOLO can only run on Python version 3, it was not possible to allow inference of live video data from the drone. Python version 3 is generally not backward compatible with previous versions, so the H.264 decoder required by the Ryze Tello EDU SDK was not compatible with the existing infrastructure. Because of this, it was decided to simplify the problem by inferring images from a webcam instead of images from the drone.

V. RESULTS AND EVALUATION

SSD reached an mAP score of 83.1%, while YOLO scores almost equally high with 80.4%. To minimize double detected objects the confidence score for SSD was set to 0.7, for YOLO already 0.25 was enough. SDD struggles to detect partly hidden objects. Also detecting objects from extreme view positions was difficult, but can be improved by adding more samples for this case. For those special cases YOLO performs equally bad, for the latter even worse. SSD reliably detects objects until a three meters distance threshold, YOLOs' maximum range is only two meters. Overlit lighting conditions leads to no or wrong object

detection using the SSD or YOLO algorithm. However, all detections of both models reacted invariantly to different backgrounds or image resolutions.

Both SSD and YOLO are capable to run inference in the real time scenario of *SmartWarehouse*, SSD with 30 PFS and YOLO with 28 FPS.

VI. Discussion

Use of the object detectors in the industry. In terms of precision, both object detectors turn out better than the original reference results from the scientific publications. However, it should be noted that the results do not compare well with those of the scientific publications, since they were trained on different, significantly simpler data. With 83.1%, SSD300 is 8.8% better, while YOLOv3 even shows an improvement of 16.7%. Nevertheless, it can be concluded that both implementations scale in terms of precision in an order of magnitude that comes after the level of the comparison results from PascalVOC 2007.

Another evaluation criterion is the responsiveness. Here, SSD300 scores slightly better with an average of 30 FPS than YOLOv3 with 28 FPS, although this difference should hardly be seen as a true decision criterion to prefer SSD300 over YOLOv3.

Regarding inference behavior both object detectors struggle with difficult lightning conditions, extreme viewing positions, longer distances and occlusions. Some of those problems could be tackled better by enhancing the dataset with additional instances for those cases.

Overall, it can be concluded that both SSD300 and YOLOv3 are object detectors for industrial use. If many objects of different scales are to be detected in the detection environment, SSD300 is preferable to YOLOv3. However, if this is not the case and more emphasis is placed on accuracy for simple datasets such as *SmartWarehouse*, YOLOv3 is the better choice.

Feasibility of the *SmartWarehouse* scenario. For reasons of extensibility of the *SmartWarehouse* scenario, YOLOv3 was selected as the final object detector. At the moment, the scenario is only designed for beverage bags as an example. If we return to the actual idea of performing inventories in large department stores with many different products, the aspect of better precision of detection for the same objects of different scales tends to fall into the background with SSD300. Rather, it is important to enable reliable detection behavior.

Still three issues remain unsolved:

- Inferring the video stream of the drone, due to the incompatibility of the runtime environments for the H.264 decoder and the deep learning models,
- The detection of occluded objects, and
- The unique counting of objects during the inventory.

The *SmartWarehouse* scenario is therefore not feasible according to the decisions made and with the given framework conditions.

VII. Conclusion

The goal of this work was to evaluate how well existing object detectors are basically suited for industrial application scenarios, and to determine whether the specific application scenario for conducting an inventory of department stores using a drone can be implemented as a prototype.

Even if object detectors such as SSD and YOLO most certainly have the potential to be used industrially, the *SmartWarehouse* scenario still could not be realized end-to-end. To do so, one would need to enhance the dataset to cover certain image capturing conditions as described, implement a custom H.264 decoder to connect the drone video stream to the selected deep learning model for real time inference and find a solution to uniquely identify objects during the detection process like additionaly making use of RFID technology.

References

[1] doks. innovation GmbH, inventairy xl (2020).
URL https://doks-innovation.com/solutions/inventairy-xl

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, Ssd: Single shot multibox detector.
URL https://arxiv.org/pdf/1512.02325.pdf

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You only look once: Unified, real-time object detection.
URL https://arxiv.org/pdf/1506.02640.pdf

[4] A. F. Joseph Redmon, Yolov3: An incremental improvement.
URL https://arxiv.org/pdf/1804.02767.pdf

[5] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, Mask r-cnn.
URL https://arxiv.org/pdf/1703.06870.pdf