2018 17th IEEE International Conference on Machine Learning and Applications

Application of Open-Source Deep Neural Networks for Object Detection in Industrial Environments

Christian Poss
Logistics Robotics
BMW Group
Munich, Germany
Christian.Poss@bmw.de

Olimjon Ibragimov
Logistics Robotics
BMW Group
Munich, Germany
Olimjon.Ibragimov@bmw.de

Anoshan Indreswaran
Logistics Robotics
BMW Group
Munich, Germany
Anoshan.Indreswaran@bmw.de

Nils Gutsche
Logistics Robotics
BMW Group
Munich, Germany
Nils.Gutsche@bmwgroup.com

Dr. Thomas Irrenhauser
Logistics Robotics
BMW Group
Munich, Germany
Thomas.Irrenhauser@bmwgroup.com

Marco Prueglmeier
Innovation and Industry 4.0
BMW Group
Munich, Germany
Marco.Prueglmeier@bmw.de

Prof. Dr. Daniel Goehring
Dahlem Center for Machine Learning and Robotics
Freie Universitaet Berlin
Berlin, Germany
Daniel.Goehring@fu-berlin.de

***Abstract*—Due to dynamics, flexibility and diversity in logistics, perception-controlled, intelligent robots are required to automate logistical handling steps. Due to the additional optical influences of the industrial environment, such as labeling or damage, these applications seem predestined for the use of generalizing deep neural networks (DNN). These showed continuous improvements over the last few years based on publicly available data sets. If these DNNs are re-trained based on training data from the industrial environment, a lower performance can be observed. The additional extension of the experiments to international locations of the vehicle plants also showed that a drop in performance can be observed in the implementation of a network trained in Germany, for example, when it is used in America. However, in order to be able to use such robots in the logistic processes in the future, further measures such as a revised composition of training data or their extension by data augmentation are proposed.**

***Index Terms*—Object Detection, Deep Neural Networks, Industrial Environment, Logistics Robotics**

I. Introduction

Currently, the automotive industry is mainly characterized by the strong rise in available part variants, a drop in vertical manufacturing, and a globally growing supplier network. On the one hand this has enabled a solid market position in the volatile and highly competitive market, but on the other hand it has lead to an increased complexity in the supply chain. This has resulted in the logistics costs of a single part to be higher than the pure production costs (15,5 % versus 24 %). [6]

One way to reduce these costs is the holistic automation of logistics processes. In comparison to the processes in the building a car, logistics processes are characterized by a flexible and dynamic environment with a very few standardization [7]. The implementation of classical automation technology is therefore not feasible. However, autonomous and intelligent robots which can adapt their tasks to the changing environments may be used. What this means for the specific application in an industrial environment, and the challenges involved are shown in the paper using objection detection as the example.

II. Actual State Analysis

Before focusing on algorithms for intelligent object-detection, a deeper insight of the material flow in a high-variant assembly line is given in this chapter. Subsequently, the relevant objects - the container of the goods in the plant - are analyzed in detail.

A. Material flow in high-variant assemblies

Logistics delivery concepts are divided into production-synchronous and production-asynchronous approaches. In the production-synchronous supply chain the parts, specially the more expensive parts, are directly delivered in the right sequence to the assembly line. In production-asynchronous process the parts are delivered through various separations, storages (AKL) and order pickings (SUMA). Also the majority of the necessary parts are still transported production-asynchronous. This process chain is visualized in figure 1.

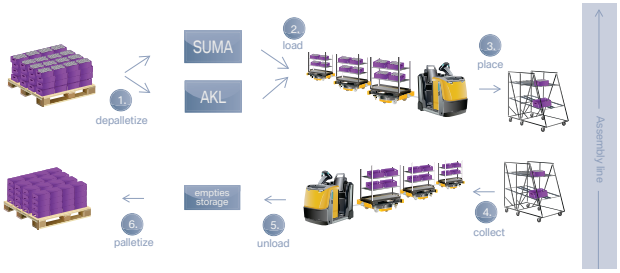

Fig. 1: Logistics Process Chain

In the production asynchronous approach, four completely manual handling steps remain, if the transportation processes

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00041

and already automated handling steps, like outsourcing containers from automated storages and the picking of single parts are not considered. They are

- depalletizing full containers,
- providing full containers,
- collecting empty containers and
- palletizing empty containers.

In the perspective of automating these processes, the handling steps vary mainly with regard to the different possible degrees of freedom and different states of the containers. The tasks and requirements for a holistic object-detection in the material flow are discussed in the next section.

B. Container and container-characteristics in industrial environments

The selection of containers takes place in the logistics planning. A safe transport without any damages on the single parts, as well as a very high packing density to reduce delivery costs are the most important planning aspects. This leads to a huge diversity of different containers in the plants. Figure 2 shows a typical example from the BMW Group's plants in Leipzig and Spartanburg. There are about ten containers with a high share on the whole material flow, and the rest of the approximate 400 container types have a share below 1 %. Additionally, the comparison of the two plants shows that depending on the cars that are being produced (small cars in Leipzig, bigger ones in Spartanburg), and on the regional placement of the production sites, the appearance of different container types can be differentiated even more. A short overview of different container types is given in figure 3.

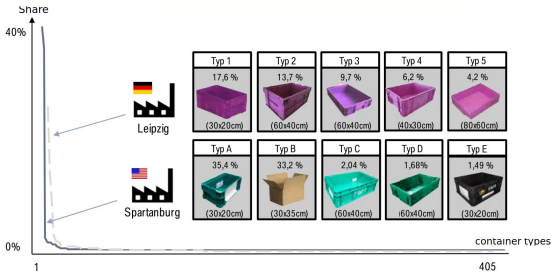

Fig. 2: Distribution of container variants in the material flow

In addition to the diversity, further deviations on the optics of the containers among the different plants can be noticed. Major influences are labels, dirt, damages and different lightening conditions. Those impacts are also shown in figure 4.

From the logistics processes, the objects, and the external influences from the industrial environment, the requirements for the object-detection algorithm and the application of the robots can be derived.

III. Requirements for Object Detection System

The basic task of object-detection in an industrial environment is the detection of containers that are placed in shelves

Fig. 3: Diversity of Containers

Fig. 4: Visual appearance of one container type in the logistics environment

or on pallets. For deploying this solution to all automotive production lines, the algorithm should be more than 99 % reliable (A).

The location information of the containers from the detection is used by the robot to determine the specific gripping point to fetch the containers one at a time. The quality of the calculated gripping points is therefore dependent on the detection accuracy of the containers. Therefore, the object detection has to be as precise as possible (B).

In every automotive plant many different cars are built that rely on different development cycles. Almost every week new parts and new containers are added to the plants. Any application for object detection has to be independent of those continuous changes to reduce the amount of manually adapting the applications to cope with the new additions and changes (C).

The robotic systems are subjected to strict cycle times. Those can vary depending on the plant and the process between 30 and 60 seconds for a complete handling step. The majority of this available time is needed for the physical movements. Accordingly the necessary time for object detection should be as short as possible (1 second) (D).

Next to those functional requirements, the functionality of the algorithms under different influences of the industrial environment has to be guaranteed. Therefore, the application should be function independent of factors mentioned below (E).

- dirt,
- lightning conditions,

- restricted field of views,
- labels and
- changing environments

IV. State of the Art

Considering the requirements and the current state of the art, the application of deep learning is the ideal choice for the problem.

A. Object Detection via Deep Neural Networks

Availability of data and acceleration in computing power allows to quickly train different neural networks. Even more importantly, the development of learning architectures was an essential push in performance. In particular, since AlexNet won the ILSVRC challenge in 2012 by a large margin, convolutional neural networks (CNNs) became the state-of-the-art approach for feature extraction from images. However, ILSVRC is not the only challenge where the most efficient and robust solutions emerged, also the other competitions shows in (Table I) have created a platform for novel solutions for object detection [8] [9] [10] [11] [12].

TABLE I: Object Detection Challenges

Name	No. of Images	No. of Classes
ILSVRC	450k	1000
MS COCO	120k	80
Pascal VOC	12k	20
CIFAR-10	60k	10
KITTI Vision	7k	3

Microsoft's COCO dataset and Pascal VOC dataset are also widely used by the open-source community to train the networks for detection of common objects like person, table, and so on. Nevertheless the use of these solutions for industrial applications are constrained by the limited number of training/testing samples and the dynamic nature of the problem.

Figure 5 shows some of the current best performing open source architectures using mean average precision (mAP) a widely accepted metric for evaluating object detection algorithms in the above mentioned challenges and the scientific community. The figure also gives information regarding real time performance through the metric of frames per second (FPS). Using these information Faster R-CNN (Region-based Convolutional Neural Networks), YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) were chosen from the detectors shown in the figure. Recently published RetinaNet algorithm was also added to the experiment due to its unique approach towards solving the problem of imbalanced training data.

The chosen architectures can be divided into two categories:

- Two-stage detectors: Faster R-CNN
- One-stage detectors: YOLO, SSD, RetinaNet

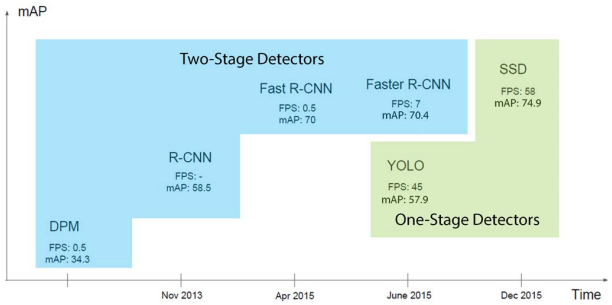

Fig. 5: Comparison of Algorithms (Pascal VOC)

B. Two-stage Detectors

In the first stage of detection, such algorithms generate a sparse set of candidate object locations, and in the second stage they classify each candidate location as one of the foreground classes or as background using a CNN.

In case of Faster R-CNN, proposals are generated with Region Proposal Network (RPN). (Figure 6)

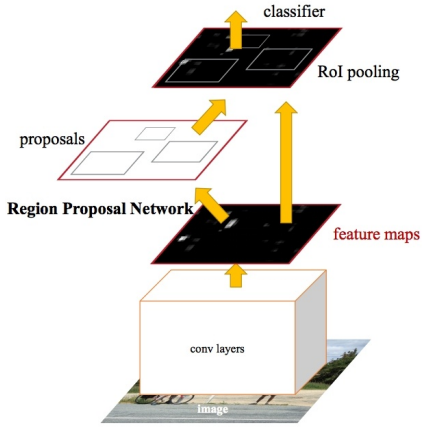

Fig. 6: Faster R-CNN Architecture

RPN is a small ConvNet (3x3 conv - 1x1 conv - 1x1 conv) looking at the conv5_3 global feature volume in the sliding window fashion. Each sliding window has 9 prior boxes that relative to its receptive field (3 scales x 3 aspect ratios). RPN does bounding box regression and box confidence scoring for each prior box. The whole pipeline is trainable by combining the loss of box regression, box confidence scoring, and object classification into one common global objective function [2].

C. One-stage Detectors

As a further advancement of two-stage detectors, one-stage detectors (also called single shot detectors) took over the performance charts by their robustness (frames per second) and accuracy rates relative to state-of-the-art two-stage methods (Figure 5).

YOLO models object detection as a regression problem for bounding boxes and object class probabilities. It applies a single pass through the CNN by dividing the input image into a 7x7 grid. Each cell predicts a distribution over class labels as well as a bounding box for the object whose center falls into it (Figure 7). [3]

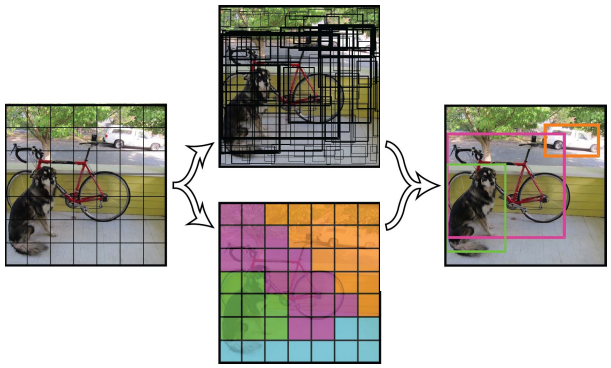

Fig. 7: You Only Look Once. Source: [3]

SSD takes advantage of the Faster R-CNNs RPN. It is used to classify an object inside each prior box instead of just scoring the object confidence (similar to YOLO). The diversity of prior boxes resolutions is improved by running the RPN on multiple conv layers at different depth levels [4].

RetinaNet addresses the problem with class imbalance as the primary obstacle which prevents one-stage object detectors from surpassing two-stage methods (like Faster R-CNN). The focal loss is applied to modulate the cross entropy loss in order to focus learning on hard examples and reduce the weight of the numerous easy negatives. Ultimately, it is a fully convolutional one-stage detector [5].

D. Evaluation KPI

Evaluation of the trained models is done by comparing the detection results with the ground truth bounding boxes, which results in calculating the mean average precision (mAP). mAP for a set of queries (in this case, set of images) is the mean of the average precision scores for each query (in this case, set of predictions with confidence levels).

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{1}$$

where Q is a number of queries.

Average Precision (AP), in turn, represents the average of maximum precision values over all correct detections. In other words, it is the area under the precision-recall curve.

$$AP = \sum precision(x) * (recall(x) - recall(x-1)) \tag{2}$$

where *precision* is an interpolated precision.

Precision is calculated as a fraction of ground truth objects from all detected objects:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Whereas, Recall is the ratio of correctly detected objects to all ground truth objects:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

In this paper, it is proposed to calculate the precision and recall values for each image. Plotting (Figure **??**) the results of each image in the testing dataset will give us Precision-Recall Curve. Then the AP value is obtained according to Equation 2.

V. Application in Industrial Environment and Evaluation

The application of the afore mentioned algorithms is now focused in the following chapter. After a short description of the training process we give attention to comparing the results with those reached at the challenges referencing above.

A. Training and Evaluation

Training data consists of 2000 images with various number of containers per images. The distribution of container types is regarding to their appearance in the material flow as shown in figure 2 as follows:

- Pink: 14.762
- Black: 6.935
- Blue: 1.603
- Brown: 190
- Grey: 185
- White: 46
- Green: 23

The neural networks were trained with the set of hyperparameters shown in Table II on NVidia GPUs (Table III).

TABLE II: Training Parameters

Algorithm	Learning Rate	Weight Decay
YOLO	0.0001	0.0005
SSD	0.000099	0.0005
Faster R-CNN	0.0001	-
RetinaNet	0.0001	0.000005

TABLE III: Hardware Configuration

Algorithm	GPU
YOLO	2x GeForce GTX 1080
SSD	2x GeForce GTX 1080 Ti
Faster R-CNN	2x GeForce GTX 1080
RetinaNet	2x GeForce GTX 1080

In turn, testing data was separated based on the location where the images were taken. This allows to evaluate the performance of the detection algorithms with regard on the frequency of particular containers in given plant locations:

- Germany: Munich, Leipzig and Regensburg
- USA: Spartanburg.

B. Results compared to challenges

As emphasized in section III, the goal is to have a system which is robust, precise, scalable, and fast. In this paper, the above mentioned parameters, except scalability, of the state of the art deep learning models are evaluated. These properties can be quantitatively analyzed through mean average precision (mAP) value for robustness, intersection over union (IoU) value for precision, and computation time for speed.

For quantifying robustness mAP was chosen as the score encloses the performance of a model on both classification and detection as already elaborated in section IV-D. From the table IV it can be seen that in the test set prepared from German plants SSD yielded the highest mAP of 61.3% and YOLO follows immediately with a value of 58.8%. Faster RCNN and Retina Net reached an mAP score of 47.3% and 35.6%. The performance differences between the models on BMW dataset also varies as that of the public data sets. On the BMW dataset the models perform better than on the MS COCO and worse than that on Pascal VOC. From the differences in the performance of the models on the Pascal VOC, MS COCO and BMW dataset, it can be noted that the preparation of an appropriate dataset is vital in reaching the necessary accuracy for industrial implementation.

To evaluate the accuracy of bounding boxes the IoU values applied in mAP calculation was used, as it quantifies the percentage of correct region in the predicted box and the ground truth. For this experiment, a prediction is considered true positive if the IoU value is greater than 40%; only the true positives were used in evaluating the bounding box accuracy. Compared to the evaluation of robustness of the models, the results of the accuracy of the bounding box predictions between the different models are similar. Retina Net performs the best with an average IoU of 69.5%, and Faster RCNN has an accuracy of 68.1% and YOLO 67.2%. The accuracy of the bounding boxes predicted by SSD is also close to the other models with a value of 60.6%. All the models have comparable performance,

The average inference time was calculated for every model and YOLO is 20 times faster than Retina Net with an average computation time of 43 milliseconds for a prediction. Retina Net takes the longest with 1056.3 milliseconds, Faster RCNN 527.9 milliseconds and SSD 491 milliseconds. However, the comparison has to consider that image passed to Retina Net is of size and to YOLO is of size 416×416. Faster RCNN uses an image of size and SSD an image of size 600×600. Also the YOLO implementation used was written on C, while the implementations of the other models were on python. The difference in the implementation language has a major role in the lower inference time for YOLO compared to the other models.

For any of the models to be used in the industrial environment both the robustness and the accuracy has to increase significantly. An alternative would be to have additional validation steps to remove false positives in the prediction and resize the bounding boxes using prior knowledge to ensure an application which is suitable of industrial use.

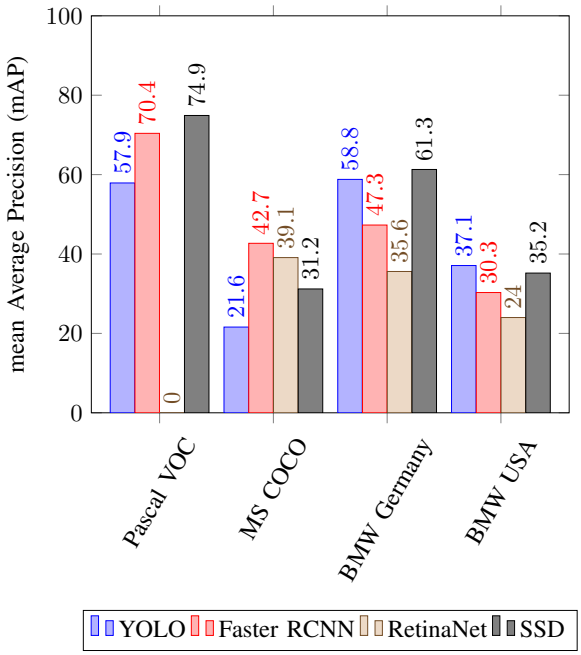

Fig. 8: Mean average precision (mAP) of state of the art models on public dataset and BMW Groups dataset. The results on the Pascal VOC dataset and MS COCO are as reported by Wei Liu et al. and Tsung-Yi Lin et al. respectively [4], [5]

TABLE IV: Performance of the state of the art deep learning models in object detection for industrial use. The mAP of the public datasets were obtained from the SSD

Country	Deep Learning Models	mAP in %	IOU in %		Detection time in milliseconds	
			Mean	SD	Mean	SD
Germany	YOLO	58.8	67.2	14.3	43.0	6.0
	Faster RCNN	47.3	68.1	13.3	527.9	75.1
	Retina Net	35.6	69.5	15.1	1056.3	289.5
	SSD	61.3	60.6	13.1	491.0	6.0
USA	YOLO	37.1	60.3	13.1	40.5	4.3
	Faster RCNN	30.3	62.9	13.9	667.4	69.8
	Retina Net	24.0	57.8	13.4	1.2686	105.6
	SSD	35.2	59.5	12.5	626.0	4.6

C. Results compared to changing environments

Generalization is one of the most important aspects for the assessment of the performance of neural networks. Usually, a model is expected to generalize well to unseen data that differ from the samples it was trained on.

For the use case at BMW Group, the main focus was on getting images from German plants to train the above mentioned models, as this is where the deployment of object detection first occurs. However, in the long run a world-wide

roll-out of current development is planned. Therefore, the test set contains images from both German and US plants. The comparison of the results of all models for the two different test sets are visualized in Figure 8. The performance measures for test set 2 (USA) demonstrate that the detection quality for all models is significantly lower than for test set 1 (GER). For example, YOLO performance is 21,7% worse for USA then for GER, while the difference for SSD performance is only 13%. On average, the difference in performance between test set 1 and 2 is 15,8%. Having a closer look on the inference results emphasizes that the models detect less containers while producing a higher false-positive rate.

There is a couple of characteristics that differentiate USA images from the ones taken in German plants. One of these characteristics is the existence of a greater variety of containers, thus increasing the generalization challenge for the models. Furthermore, the whole environment is looking differently, although the focus is still on the automotive production plants only. For example, on average, the containers are covered with more stickers, which occlude the visible surface of the containers and are therefore increasing identification difficulty.

As introduced in section A, the training set consists mainly of images from German plants. Images from US-plants are underrepresented. Taking in consideration the different characteristics between the plants in Germany and the USA - and therefore the differences between the training set and test set 2- the drop in detection quality seems plausible.

In general, the results are not as good as required. The main reason is that our data base is insufficient. We need more data and our data set is not balanced enough. Therefore, our models are not able to generalize and perform as well as we would need it to do.

VI. Future Work

The comparison of the different neural networks on public datasets and on manually labeled industrial datasets shows a big gap regarding the evaluation kpi. On average a discrepancy of 20 % can be identified. With respect to the aforementioned requirements for deploying neural networks to an industrial environment, we have to state that the application of open-source algorithms is not yielding satisfying results.

Even when considering the performance of the models on public datasets, an application in the logistics process chain seems unfeasible. The robots have to deal with up to 20k containers per day. Therefore, a performance error of 20 % would lead to 4k human interventions, which is way too expensive for such cost-intensive fields of application.

For our future work, we aim for improving the data base as well as the optimization of hyperparameters for our learning algorithms. Following our goal to deploy our work to every plant in the BMW network, we define the following measures for further research:

1) Increase the number of training data
2) Define equally distributed training set (which represents all plants and their characteristics).
3) Train a different model for each country, thus taking into account all the plant-specific characteristics while keeping the overall complexity at a low level.
4) Introduce new object (sticker/label) and implement logical checks (e.g.: if the model detects a label, there should also be a container).

In order to improve the datasets finally a deeper insight in the different influences is important. Not only the distributions of the different container types also the distributions for examples of the number of labels per container could matter. This is the outline for future research.

References

[1] J. Philbin et al, “Object retrieval with large vocabularies and fast spatial matching”, Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on, pp. 1–8, 2007.

[2] S. Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, CoRR, abs/1506.01497, 2015.

[3] Joseph Redmon et al, “You Only Look Once: Unified, Real-Time Object Detection”, CoRR, abs/1506.02640, 2015.

[4] W. Liu et al, “SSD: Single Shot MultiBox Detector”, CoRR, abs/1512.02325, 2015.

[5] T. Lin et al, “Focal Loss for Dense Object Detection”, CoRR, abs/1708.02002, 2017.

[6] W. Guenthner, ”Effiziente Logistik - zentraler Erfolgsfaktor”, fml-Lehrstuhl fuer Foerdertechnik Materialfluss Logistik, TU Muenchen, 2011.

[7] D. Arnold et al, ”Handbuch Logistik”, Springer-Verlag Berlin Heidelberg, 978-3-540-72929-7, 2008.

[8] A. Karpathy et al, ”ImageNet Large Scale Visual Recognition Challenge”, abs/1409.0575, 2015.

[9] T. Lin et al, ”Microsoft COCO: Common objects in context”, abs/1405.0312, 2014.

[10] M. Everingham et al, ”The PASCAL Visual Object Classes (VOC) Challenge”, IJCV, 2010.

[11] A. Krizhevsky et al, ”The CIFAR-10 dataset”, online: http://www.cs.toronto.edu/kriz/cifar.html, 2014.

[12] A. Geiger et al, ”The KITTI vision benchmark suite”, online: http://www.cvlibs.net/datasets/kitti/evalobject.php, 2017.