Comparing Halcon and Detectron2 for detecting impurities in pig intestines used as natural sausage casings

Author: Aske Bach Lassen, aala@teknologisk.dk, +45 72 20 35 79
Affiliation:
Danish Technological Institute,
Forskerparken 10F,
5230 Odense
Denmark
www.dti.dk

Abstract

Pig intestines used as natural sausage casings are currently cleaned and then sent from Europe to China for manual quality control and grading. This process can be made greener by automation that removes the need for transport. A suitable machine now exists, able to check the casings for leaks and to grade and cut them to standard lengths. The one remaining quality control process is checking for impurities left by the cleaning process. A sophisticated vision system and a deep learning system is needed.

After preliminary lighting tests, images of cleaned pig intestines destined for sausage casings were examined manually for impurities. Pixels depicting impurities were labelled and mask impurity images produced as "ground truth". Two deep learning methods were applied in order to predict the areas of impurities in these images: Halcon with SOLOv2 semantic segmentation and Detectron2 with Mask-R-CNN instance segmentation. Despite the over-abundance of background pixels, both algorithms learned the segmentation. Detectron2 was more accurate but Halcon found more of the impurities, which was attributed to the difference between the segmentation types: semantic vs instance. Since the aim of this work is to produce clean intestines for use in the food industry, false positives are more acceptable than false negatives, so Halcon was chosen.

Keywords: Halcon, Detectron2, sausage casings, quality control, detection of impurities

Introduction

Intestines are often used as natural sausage casings, that is as edible containers for various types of sausage. They are therefore subject to standard food safety and hygiene requirements. The current processing system involves extracting and cleaning the intestines at the local abattoir and then shipping the intestines to low-wage countries such as China for quality control, grading and cutting. This involve filling the intestine with water, measuring the intestine diameter and thickness, looking for any leaks (caused by holes) and looking for "impurities" (remains of faeces). Where holes or impurities are found that section is cut out and discarded. This process occurs manually at high speeds. Due to the transportation time, sausage casings are around 6 months old before they are filled and the sausages sold as human food. It would be better to perform the sorting more locally, making the product fresher, reducing the CO_2 emissions associated with the transport, and increasing traceability. It is already possible to automatically detect holes, grade and cut sausage casings. This project investigates how to automate the final step of detecting impurities in pig intestines and thereby enable moving the production of food-quality natural sausage casings back to Europe.

SelectiCa is a three-in-one robot in the final stages of development (Proxima Centauri, 2020) that will be able to perform the sorting and grading tasks to at least the standard obtained by manual workers. It can automatically pick the casings up, fill them with water, check for leaks and cut where a leak is detected, measure the diameter and sort into standard sizes, at 2m/s and without any direct labour required. Most of the grading and quality control in SelectiCa is performed by a 3D vision system that scans the surface of the water-filled casing. From this, precise diameter measurements can be performed, as well as holes where water is spraying out, but impurities are harder to detect. Impurities vary considerably in size, shape, intensity etc., and often look a lot like elements of the intestines that are not impurities. This makes the correct classification of impurities a challenging task and therefore makes deep learning the method of choice, as it can find quite subtle patterns if given sufficient annotated data. Here, we use a convolution neural network (CNN) architecture for our deep learning process.

Region-based CNN (R-CNN) methods create boundaries around every object that is present in the image. They divide the image into regions and use a greedy algorithm to recursively combine similar regions. The proposed regions are fed into a CNN which produces a high-dimensional feature vector for each, plus offset values to improve the precision of the bounding box. This process takes much too much time for real-time applications, so faster methods have been developed.

Detectron2 (Wu et al., 2019) was built by Facebook AI Research (FAIR) to support rapid implementation and evaluation of novel computer vision research. It is based on Pytorch, an open-source optimized tensor library for deep learning based in Lua but with a good Python interface and reasonable C++ interface. Detectron2 is very easy to use and supports multiple advanced deep-learning methods. For this case, we want a method with high accuracy rather than high speed. We also prefer to use established algorithms to maximize stability, reliability, and robustness for this industrial application.

We therefore chose Mask R-CNN (He et al., 2017), which is an instance segmentation method also developed by FAIR that identifies every pixel of every object in the image (instead of only using bounding boxes). It uses a backbone CNN network supplemented by a feature pyramid network that is better at representing objects at multiple scales. It involves using small masks to compute losses and then scaling the masks up to the size of the bounding box during inferencing. This region of interest (ROI) pooling enables sub-pixel accuracy.

In this paper we investigate how well the Detectron2 Mask R-CNN method detects impurities in pig intestines based on images obtained using an industrial 2D camera integrated with the SelectiCa robot. We will compare this to the impurities detected by the commercial system *Halcon* (MVTec). Halcon software is often used for research-level machine vision applications and is certainly very flexible and has a good development environment, but it is rather expensive for industrial use. It does not (yet) support instance segmentation but is very strong for semantic segmentation (where objects are found and labelled by class). Since for this application we are only concerned with a binary classification of whether sections of casings contain impurities or not, semantic segmentation should be sufficient.

In order to use the same evaluation method, we will base our assessment of both methods on evaluation algorithms developed in the Halcon framework. We therefore ran standard Halcon segmentation and object detection on our images and compared this Halcon output with the output from Detectron2 running Mask R-CNN to discover which is better. The aim of this work is to make a vision solution that would work as a future product upgrade for Proxima, so that they can take commercial advantage of the results derived from our research.

Method

The hardware we used for obtaining images and training the networks, apart from the latest SelectiCa prototype, was the following:

- Grayscale camera: Basler acA2440-20gm, resolution 2440 x 2048 pixels
- Colour camera: Basler acA1920-155uc, resolution 1920 x 155 pixels
- Custom-made RGB LED diodes giving the possibility of specific light wave lengths, located in the mandrel
- Back light: MBJ Imaging DBL3030-WT with white LEDs (5000K)

Lighting tests

Correct lighting is a very important factor for computer vision tasks. Initial investigations into lighting involved multiple lighting tests to come up with a solution that was optimal for the harsh food-grade environment it was going to work in.

Different lighting set-ups were tried before the main tests started. Front lighting creates too much glare, as the intestines are very wet and therefore reflective. Back lighting creates a more muted and diffuse light where impurities show up well as darker patches. Blood vessels, connective tissue, etc. mostly show up as paler tracks, sometimes with a shadow.

Figure 1: Grayscale image of a water-filled pig intestine in the SelectiCa machine, lit by back lighting, showing a fountain produced by a small hole.

Another idea was to install LED lights in the mandrel that the casings slide along, so the intestine is lit from the inside. Both impurities and veins show up as lighter than the background (Figure 2).

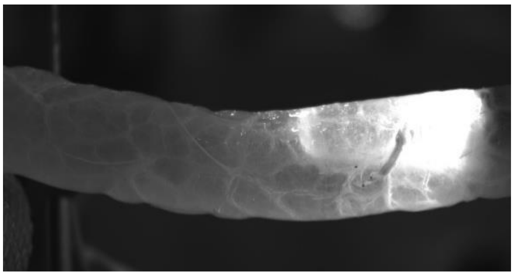


Figure 2: Grayscale image of a water-filled pig intestine in the SelectiCa machine lit from inside (LEDs in the mandrel)

Since annotating by hand is a very time-consuming task and the image background contains a lot of irrelevant information, the images are cropped to a *1101 x 901* pixel region of interest. This means that any impurities take up a larger proportion of the image, so learning is easier.

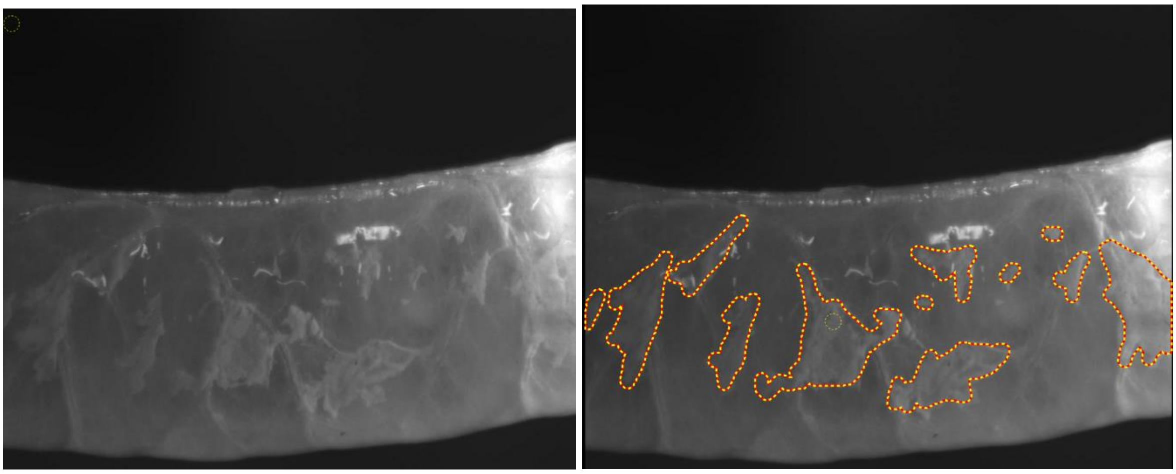


Figure 3: Left: grayscale image of a water-filled intestine. Right: same image segmented to display the impurities found.

Our investigations into lighting led to the following conclusions and decisions for the training and comparison phase:

- **Light position:** Lighting from the front does not work well due to reflections. Back lighting works well but would require considerable modification to the SelectiCa machine. Internal lights give almost as good results as back lighting and require very little modification and extra cost, so this was the lighting scheme chosen.
- **Light:** The colours amber, red, green, and blue were tested for the LEDs in the mandrel. However, using a specific colour did not improve image quality. Therefore the LED colour was chosen to be amber, which was converted to grayscale using the grayscale camera.
- **Camera:** The visual image of impurities was not improved by using light with different wavelengths or by using a colour camera. The grayscale camera has a larger dynamic range and so was the one chosen, as it performed better.

Segmentation method

It is important to identify and localize any impurities present in the pig intestine, so that these sections can be removed or rejected to ensure that those casings used for sausages comply with food hygiene requirements. Since there is no current standard for how to identify and classify intestine impurities, the choice of segmentation method and learning model is relatively open.

The main types of segmentation method suitable for this case are:

- **Object detection:** if the requirement is the number of impurities, or a relative measure of impurity area, a bounding box could be used. There exist many accurate and fast object detection models, such as YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) and many more.
- **Semantic segmentation:** if a relative pixelwise measure of how much impurity there is in each image is needed, then semantic segmentation would be the optimal choice.
- **Instance segmentation:** This gives both a bounding box and a mask for each object and is therefore the most versatile method. Since there is not yet any definitive specification of how to use the results of the segmentation of impurities, instance segmentation is preferred.

Real-time results are essential as any section of casing containing impurities must be cut out while that intestine is still on the machine and so distances to the impurities are calibrated. One of the current top-performing, real-time instance segmentation models is SOLO v2 (Wang et al., 2020). Figure 4 shows a comparative performance table from that paper.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
box-based:							
Mask R-CNN [4]	Res-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN*	Res-101-FPN	37.8	59.8	40.7	**20.5**	40.4	49.3
MaskLab+ [30]	Res-101-C4	37.3	59.8	39.6	16.9	39.9	53.5
TensorMask [7]	Res-101-FPN	37.1	59.3	39.4	17.4	39.1	51.6
YOLACT [2]	Res-101-FPN	31.2	50.6	32.8	12.1	33.3	47.1
MEInst [9]	Res-101-FPN	33.9	56.2	35.4	19.8	36.1	42.3
CenterMask [31]	Hourglass-104	34.5	56.1	36.3	16.3	37.4	48.4
BlendMask [8]	Res-101-FPN	38.4	60.7	41.3	18.2	41.5	53.3
box-free:							
PolarMask [10]	Res-101-FPN	32.1	53.7	33.1	14.7	33.8	45.3
SOLO [1]	Res-101-FPN	37.8	59.5	40.4	16.4	40.6	54.2
SOLOv2	Res-50-FPN	38.8	59.9	41.7	16.5	41.7	56.2
SOLOv2	Res-101-FPN	39.7	60.7	42.9	17.3	42.9	57.4
SOLOv2	Res-DCN-101-FPN	**41.7**	**63.2**	**45.1**	18.0	**45.0**	**61.6**

Figure 4: Table showing the performance of different instance segmentation algorithms, from Wang et al. (2020)

Hyperparameters

A critical task when setting up a high-performing neural network is selecting the best hyperparameters. For the Detection2 algorithm, a grid search was performed among these settings:

Setting	Values tested	Value chosen
Network model		Mask-RCNN: mask_rcnn_R_50_FPN_3x.yaml
Learning rate	[0.0001, 0.001]	0.001
Iterations	[1500, 2000, 5000]	2000
Batch Size	[64, 128, 256]	256
Image Size		Not adjustable

Very similar hyperparameters were tested to find the best Halcon deep learning network:

Setting	Values tested	Value chosen
Network model		pretrained_dl_segmentation_enhanced.hdl
Learning rate	[0.0001, 0.001, 0.01]	0.001
Momentum	[0.5, 0.75, 0.9, 0.95, 0.99]	0.95
Weight Decay	[0.00001, 0.0001, 0.001, 0.01]	0.01
Batch Size		225
Image Size		274x224

Annotation of images

Ground truth was obtained manually, by a researcher looking at the training and test images and hand-labelling any impurities found. We used the *DTI annotation tool* to perform this annotation and saved the impure regions as mask images as shown in Figure 5, in a format readable by both Halcon and Detectron2. Any tool capable of creating masks or polygons can be used for this task e.g. MakeSense (n.d.) which is used via a browser and LabelMe (Wada, 2016) which is used on a local platform. These mask images are used by the algorithms as training data so that the algorithm knows what to converge to.

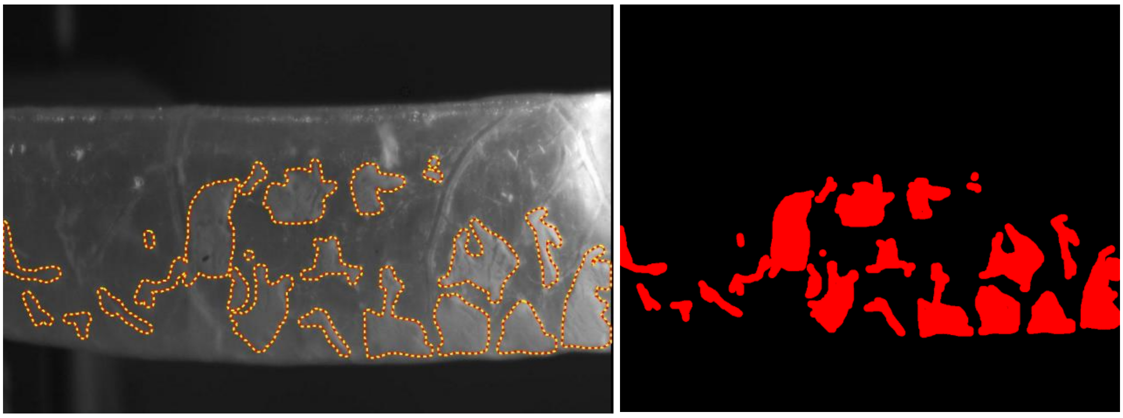


Figure 5: Annotated areas converted to binary masks using the DTI annotation tool

The Detectron2 API for Mask-RCNN takes as input a file in JavaScript object notation (ECMA, 2017) with annotation in a *common objects in context* (COCO) (Lin et al., 2014) style which can also be read by Halcon. Therefore, the mask image annotations were converted to .json format using an internally created script.

Comparing Halcon and Detectron2 outputs

Halcon as default outputs pixel accuracy and intersection over union, whereas the Detectron2 code outputs average precision (also based on intersection over union, but this parameter is hidden). Therefore, instead of comparing evaluations from each algorithm, we decided to rely on the evaluation method of Halcon. To evaluate the Detectron2 results, the predicted impurity regions from the network were injected into Halcon's evaluation procedures in order to generate results in the same format as Halcon.

Results

Initial experimentation using simplified tests confirmed that the average precision obtained from the Halcon output processed by our new code was essentially the same as that obtained using Detectron2. This also confirms that it is possible to find impurities using both semantic and instance segmentation coupled with deep learning.

The Detectron2 algorithm detects *instances* of potential impurities and marks each region containing an impurity in colour, as shown in Figure 6. The results show that the regions identified by Detectron2 as impurities are about the same size and shape as the impurities labelled as ground truth. Most impurities are found (though not the large area in light blue on LHS of the ground truth image nor a couple of smaller regions, notably the red top middle and the red bottom middle; and the green bar at the bottom is condensed to a small dark dot). One additional impurity (RHS, blue) is found that is not present in the manually-annotated ground truth image.

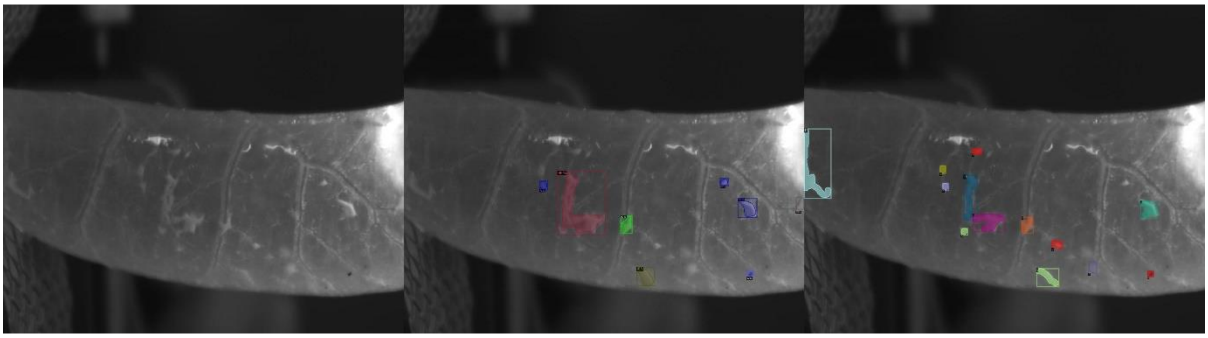


Figure 6: Detectron2: Left: original image, Centre: predictions of impurity, Right: ground truth annotation

The semantic segmentation output from Halcon is shown in Figure 7. Halcon identifies regions containing impurities, it does not attempt to identify individual instances of impurities. Not unexpectedly, we see that the regions estimated to contain impurities are considerably larger than the ground truth regions. This gives Halcon less opportunity to miss an individual impurity in a contaminated area. All ground truth impurities are found and several extra places are identified as contaminated, notably on the RHS.

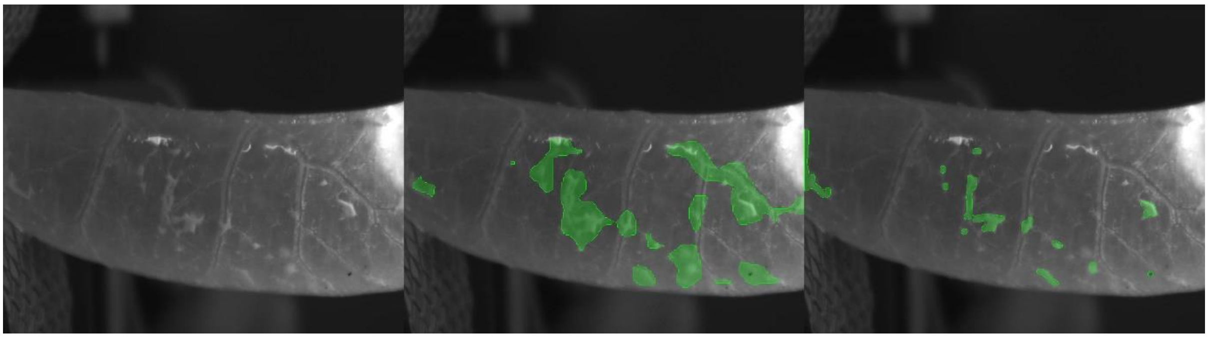


Figure 7: Halcon: Left: original image, Centre: predictions of impurity, Right: ground truth annotation

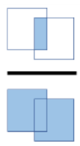
Analysis of results

The two most important measures used here are intersection over union (IoU) and pixel accuracy, as a standard bounding box method cannot be used for instance segmentation.

Intersection over union (IoU)

IoU is defined as the division of the area of overlap/intersection with the area of union, as depicted below:

$$\text{IoU} = \frac{Area\ of\ intersection\ (overlap)}{Area\ of\ union\ (total\ area)} =$$



IoU	Background	Impurity	Mean
Halcon	0.989	0.277	0.633
Detectron2	0.996	0.456	0.726

Since a very large proportion of the pixels were background, both trained networks are very good at predicting background. The IoU of pixels depicting impurities is notably lower, especially for Halcon, so Detectron2 appears better. This is unsurprising, as the regions predicted by Halcon as containing impurities are noticeably larger than the impurities themselves, so there are many false positives to increase the IoU denominator.

Pixel accuracy

Pixel accuracy is the ratio of pixels predicted with the correct class-label (true positive plus true negatives) over the total number of pixels.

Pixel Accuracy	Background	Impurity	Mean
Halcon	0.991	0.782	0.886
Detectron2	0.999	0.579	0.789

Again it can be seen that both deep learning algorithms perform very well with the background and less well with the impurities -- and here, Halcon looks better. The higher impurity detection accuracy obtained from Halcon is not surprising, given that the Halcon impurity regions contain so many extra pixels (compare the area classed as containing impurities in Figures 5 and 6) and some of these are pixels genuinely depicting impurities.

Confusion matrix

A pixel confusion matrix shows the percentage of pixels (in total) found to be background or impurity. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. The green numbers are true positives, while red ones are false positives.

Halcon pixel confusion matrix	Background	Impurity
Background	98.543 %	0.112 %
Impurity	0.941 %	0.403 %

Detectron2 pixel confusion matrix	Background	Impurity
Background	99.344 %	0.217 %
Impurity	0.140 %	0.230 %

Again, both algorithms are shown to work well on background and badly on impurities. However, these matrices also clearly show the great disparity between the number of background pixels in our tests and the number of pixels depicting impurities. This means that small errors in identifying the impurities during the determination of the ground truth data will have a large impact on the results.

Discussion

In any task involving the detection of a rare occurrence as here, the true negative “no problem” pixels greatly outnumber the problem (in our case, impurity) pixels. This means that it is easy for a learning algorithm to learn to label every pixel as negative and still be assessed as reasonably accurate. Both algorithms tested here managed to avoid this problem and detect some impurities.

The Detectron2 algorithm performs notably better on background pixels. This is not surprising, as the larger size of the Halcon “impurity” regions mean that they necessarily include a number of incorrectly labelled background pixels. The extra area covered by the Halcon-predicted “impurity” class means that more genuine impurities are correctly labelled AND more background pixels are falsely labelled as depicting impurities. For similar reasons, the Detectron2 algorithm identifies fewer of the true positives AND is better at correctly labelling background pixels. Impurities often occur in small groups (as in the image shown in Figures 5 and 6) so Halcon’s extra area may often accidentally overlap with other true impurities. Detectron2’s parsimonious labelling can also cause occasional pixels to be missed around the edges of an impurity. These comments are likely to be correct for all semantic vs instance segmentation of these images.

We find that both algorithms work well on detecting areas containing impurities in the image stream of casings moving past the camera, even at 2m/s. The Detectron2 neural network is more accurate in its classification so is more satisfying to researchers, but it can be argued that it is more important to detect ALL faecal contamination than to be accurate. Some level of false positives and the resulting loss of clean intestine is better than missing genuine impurities and them ending up in food. This argument would mean that the Halcon results would be better in a commercial, food production setting. Whether Halcon with SOLOv2 is the best of the semantic segmentation models, and whether Detectron2 with Mask-R-CNN is the best of the instance segmentation models, is unknown as yet. It would be interesting to run similar tests with other segmentation methods e.g. from the Halcon library, as future work.

If these detections methods yield good results, Proxima will consider offering impurity detection as an add-on to their current system.

Acknowledgements

This work took place under the Robot Union project funded by the European Commission under grant agreement ID: 779967. Access to the SelectiCa robot was facilitated by the developers, the company Proxima Centauri.

References

1. He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017. Mask R-CNN. Retrieved June 2020 from https://arxiv.org/abs/1703.06870 .
2. ECMA 404 2017. The JSON Data Interchange Syntax. https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf .
3. Lin, T-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C.L. 2014. Microsoft COCO: common objects in context. CoRR, abs/1405.0312.
4. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. 2016. SSD: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham. doi:10.1007/978-3-319-46448-0_2 .
5. Make Sense n.d.. Retrieved June 2020 from https://www.makesense.ai .
6. MVTec Software GmbH n.d.. Halcon -- the power of machine vision. Retrieved Nov 2020 from https://www.mvtec.com/products/halcon .
7. Proxima Centauri SelectiCa video 2020. Retrieved Jan 2021 from http://proxima-centauri.dk/wp-content/uploads/2021/01/Promo_SelectiCaSorterC1.mp4
8. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). DOI: 10.1109/CVPR.2016.91 .
9. Wada, K., 2016. Label Me: Image Polygonal Annotation with Python. Retrieved June 2020 from https://github.com/wkentaro/labelme .
10. Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C. 2020. SOLOv2: Dynamic and fast instance segmentation. *Advances in Neural Information Processing Systems*, *33*. https://arxiv.org/pdf/2003.10152.pdf .
11. Wu, Y., Kirillov, A., Massa, F., Lo, W. Y., & Girshick, R. 2019. Detectron2 is FAIR's next-generation platform for object detection and segmentation.