2023

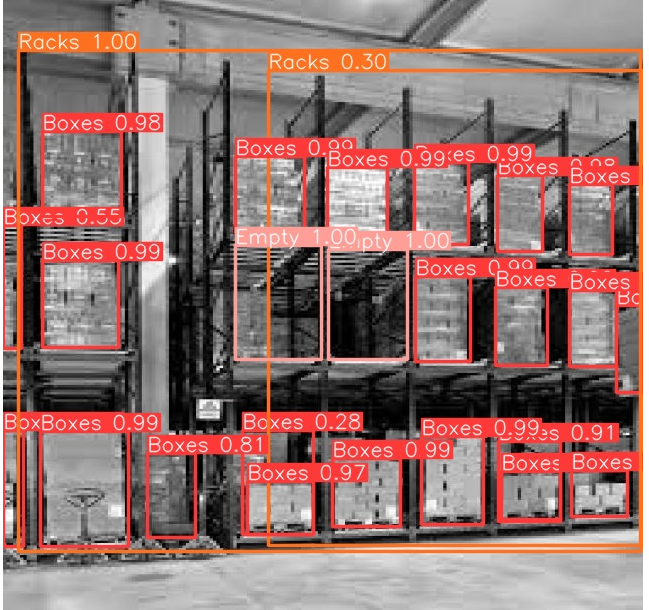
Robert-Jan Roest

Han university

19-12-2023

Individual Project

The usage of AI for the inventory of ITEQ Industries



**Title:**  Individual Project

**Sub-title:** The usage of AI for the inventory of ITEQ Industries

**Students:** Robert-Jan Roest

2136594

E: [RJ.Roest@student.han.nl](mailto:RJ.Roest@student.han.nl)

**Study Backgrounds:** Industrial Engineering and Management

**Class:** MDD-08

**Educational Institution**:  Han University of Applied Sciences

 Ruitenberglaan 26

 6826 CC Arnhem   
 T: 024 353 0500

**Organization:** ITEQ Industires

**Lecturer:** John Smits

 E: [john](mailto:Mohsen.Ghanadzadeh@han.nl).smits@han.nl

**Date:** 19-12-2023

**Place:** Arnhem

**Version:**  1.0

**GitHub Link**

https://github.com/rjroest/IndividualReport/upload/main

**GoogeColab Link**

https://colab.research.google.com/drive/1yESlpnToZbueSfBwt4z87StT9J-WvYtT#scrollTo=t6MPjfT5NrKQ

# Introduction

During the workshop given at MPack, I got to work with a software called YOLO. YOLO is a Real-Time Object Detection software. From the moment I started working with it, I knew that I wanted to implement this software for my individual report, but how?

Then I thought about a problem I encountered a short time ago, during my work at ITEQ I spoke with a colleague who works on a reach truck, he always got the feedback from the management that he did not put the pallets with products on the right place in the rack. His response to the problem was that, during the day, he got so many pallets to route and reroute that he does not have the time to consistently see where there is place within a rack and where there is none. For the management team, having a camera that sees the empty places on the shelves could be a first step in the automation process toward automated guided vehicles.

For this reason, I choose to solve the following business problem: make it clear for the software to see, based on a picture, whether there are empty shelves in a rack.

This would give the benefit for the company that they can either choose to use a different inventory system management, use different time slots for deliveries, start using another warehouse management system or see this as a first step towards the usage of automated guided vehicles. Whatever the outcome, it should give a clear insight in the current inventory level for the racks.

The report is guided by the CRISP-DM model and the software being used is the software from YOLO which was provided to us during the workshop.

Inhoud

[Introduction 1](#_Toc153866917)

[Business understanding 3](#_Toc153866918)

[Business objectives 3](#_Toc153866919)

[Customer Profile 3](#_Toc153866920)

[Value Map 3](#_Toc153866921)

[Assess the situation 3](#_Toc153866922)

[Data Understanding 5](#_Toc153866923)

[Collect the data 5](#_Toc153866924)

[Describe the data 5](#_Toc153866925)

[Verify data quality 5](#_Toc153866926)

[Data preparation 6](#_Toc153866927)

[Data modelling 7](#_Toc153866928)

[results 7](#_Toc153866929)

[Evaluation 9](#_Toc153866930)

[Of the model 9](#_Toc153866931)

[Of the results 10](#_Toc153866932)

[Deployment 11](#_Toc153866933)

# Business understanding

## Business objectives

To make the business understand as clear as possible, the Value Proposition Canvas from Dr Alexander Osterwalder (Osterwalder, 2023) will be used. This canvas makes it really clear what the pains and the gains from the company are, and how to resolve them. In this case, the customer will be the company (ITEQ), and the company will be me, providing the product for the company.

### Customer Profile

Gains – the gains describe the benefits the customer expects and needs, these are the things that increase the likelihood of adopting the value proposition. In our case the gains would most likely be that our customer, the management team from ITEQ, gets a good understanding of the capabilities of AI within their production process.

Pains – these are the negative experiences/emotions that the customer experiences in the process of getting the job done. In our case that would be where the inventory level gets too high which leaves them with only one solution: put pallets on the production floor. This is a bad scenario because the products get damaged easily and they have to be put into the racks at a later stage.

Customer jobs – this is the task the customer tries to perform or the problem that they are trying to solve. In our assignment the customers job is the task of putting the pallets in the correct places in the racks.

### Value Map

Gain creators – how the service creates gains for the customers. The service would create gains by showing where there is still place within the racks in an easy way.

Pain relievers – a description of how the service alleviates the customers pains. By providing an easy to use, real-time visualisation of the available space for the workers on the floor.

Service – the product or service that will take away all the pains and create gains for the customer. This is a code in python that runs of photo’s provided by me. It detects the empty spaces and shows where there is a place left within the rack.

## Assess the situation

In order to succeed for this assignment, a series of things are being needed. First of all, we need a lot of photos from the racks. Since I only have a few photos, photos from the internet will be used as well. This is actually a great thing. Because I use photos form the internet, there will be things in the racks that would normally not be there. This is great for if there are alternative objects or pallets within the racks, then the software will detect those as well.   
Another thing that is being needed is a software that can read the photos, determine what is in the photo and give a certain output. For this, Google Colab is used on which the YOLO software can run. Then the photos will be imported in there and via the Roboflow software the correct labels will given to the photos. In the end, all the code an stuff will be uploaded via GitHub.

The biggest risk that comes with the project is the amount of knowledge I currently contain on the subject and the software, and the possibility that the software does not recognize the spaces in the photos. But these are challenges that we could overcome with the correct usage of our resources.

# Data Understanding

## Collect the data

Some of the data will be collected via photos made by me, the other data will be taken from the internet. Like I said before, this will only be better since we get more photos of racks being filled with different products. The data will be uploaded to the software in Roboflow and there it will be altered for future usage.

## Describe the data

The data that we use are photos from racks from all over the internet. The reason for this was mentioned above. The data will contain graphical images of racks that have either full or empty shelves, the software should start to see a pattern in the photos in which the image either has empty or full shelves. (see figure 1) This rack contains, as can bee seen, empty and full shelves.



Figure : figure of a rack

## Verify data quality

The biggest problem with the data quality is that the racks are stacked really close to eachother, getting a good picture of the racks can therefore be quite hard. Since I had to take photos with the wide-angle camera, some pictures can be hardly used, or only parts of pictures. This could become a problem when the software starts detecting things outside of the picture. Another problem that could occure is that the model does not see the difference between a box and empty. Because of double racks, an empty space does have a pallets with boxes behind it, which will maybe result in the model thinking that the empty space is filled.   
Moreover, ‘empty’ is a really vague thing for a software to understand. Because the images are in 2D, empty does not really exist, it just exists of pixels. Empty could therefore better be described as: anything but box, but withing the racks.

# Data preparation

To work with the data, I had to make certain choices. Because I did not have hundreds of photos, my data could not be evenly split in training, testing and validation data the way it should according to the way it was described by Analytics Vidhya. According to them, it should at least me 60-20-20 percentage, for me, the best option was to use all of my photos as training data and use new photos to validate them.  
Another way to get more data instantly was by making the current pictures black and grey. For the software this was a totally new picture, for me it was an easy way to overcome the problem of not having enough pictures.

During the preparation part, a problem rose that only could be seen in the result. When selecting the empty places or boxes, I noticed that they looked really alike. The big problem with the empty space is that there is always something in it. You can not really detect ‘empty’. Furthermore, because it was a place in a rack, the shape was also the same.

The data was integrated via different sources from the internet and then used for training data.

I used three types of objects within the photos:

* Racks: these are the most outer lines of the racks, by selecting only boxes and empty spaces within these lines, I hoped that the software would understand that outside the racks no empty spaces exist. Because ‘empty’ is essentially anything but a box for the software, I hoped that, by using the rack as a outline, this problem would be solved
* Boxes: boxes are basically everything that is in the rack. Thus, each filled spot appears as a box. This was done on purpose because at ITEQ each product is actually stored in a box on a pallet
* Empty: Empty is essentially everything withing the rack that is not filled in. When I selected the empty spots, I found out that most of them were darker than the boxes, so I hope that the model sees that as well and can detect the vague concept of ‘empty’ in some way.

# Data modelling

The first choice for the modelling part was the usage of the YOLO software. This software is a free to use software for students. When uploading photos, you can choose between three options to work with for your photos: Object Detection, Classification, and Instance Segmentation. Out of these three I choose the Instance Segmentation. I did this because Object Detection is purely AI based, you can not classify products by yourself, moreover it finds multiple things and their specific location. I was not looking towards that exactly because I wanted to alter the products myself. In addition to that, the boxes in the photos had various sizes and shapes, therefore the Object Detection would not work properly since the boxes had so many different sizes and shapes  
Secondly, we had Classification. Classification was not the right fit because classification assigns labels to the entire image instead of to certain products.  
Last but not least we had Instance Segmentation. Instant Segmentation was the way to go because it could detect multiple objects and their actual shape. So, boxes could be boxes, whatever the shape, and empty would be empty.

For the test design I choose to use training data only. Because I did not have very may photos, I needed them in my favour. To create more photos instantly, I used the “Greyblack” function, this function makes a copy of all of your photos and turns them black and white to use them as a new photo. This way, more photos were created instantly.

The code used for the YOLO software was provided to us by the lecturer at the MPack workshop. The code had to be altered a little bit and some choices still had to be made. To train the model, I used 30 epochs. An epoch is the complete passing through all of the dataset exactly one time. This means that the model went through all the photos exactly 30 times in order to train as much as possible. To get the right number is a little bit trial and error and won’t be right the first time.



Figure : output from the model

## results

When the first results came in, I was very pleased to see that the software could detect the differences between empty (pink) and boxes (red). Even in the middle from figure 2, the model did not see the box from the row behind it as a box, this was one of my concerns. Since the model did most of the pictures really well, I was happy with the outcome.

Other pictures had some minor flaws in them, see figure 3. The software started to see the ceiling as a box as well, other parts of the ceiling were suddenly empty. Well yes they were, but not in the way I intended😊. The usage of ‘racks’ (orange) did unfortunately not exclude the variables from moving all over the picture.

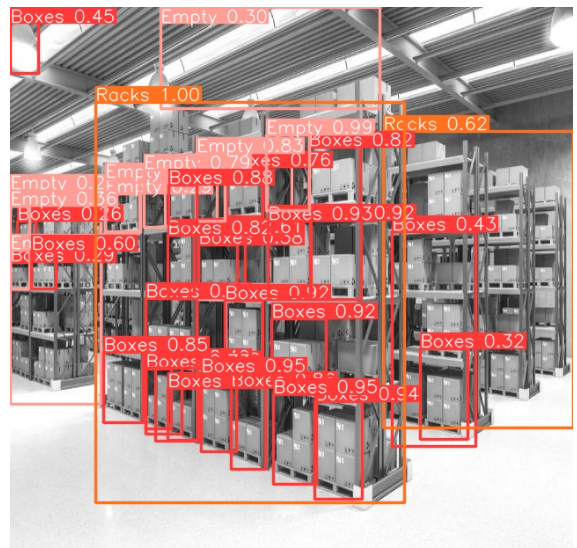


Figure : picture with some flaws

What did surprise me even more was that in picture 3, even the racks on the left were filled in with red squares, this is purely the software because I did not do those. So, despite going out of its boundaries, it did work for the most part

# Evaluation

## Of the model

After training the data for 30 times (the 30 epochs) I wanted to see if it could do the same for a new picture. I purposely did not upload one picture, because I though it was too difficult for the software to train with. But because it gave such good results I wanted to see if it could detect the empty shelves on this difficult picture. This picture is rather difficult because it was taken from further away and a lot of stuff is in front on the racks. The picture is shown in figure 4. As can be seen in the result, figure 5, it was indeed a difficult picture. It did see, funny enough, most of the empty shelves, but did not see the boxes very well. Most likely this is because the picture is taken from far way and a lot of new objects are in the picture in which the model has not been trained yet. It did see the rack correctly.



Figure 5: result validation photo



Figure 4: first validation photo

The evaluate the model, we take a look at some of the charts it provided for us. The first table we take a look at is the overview form the different epochs, runs.

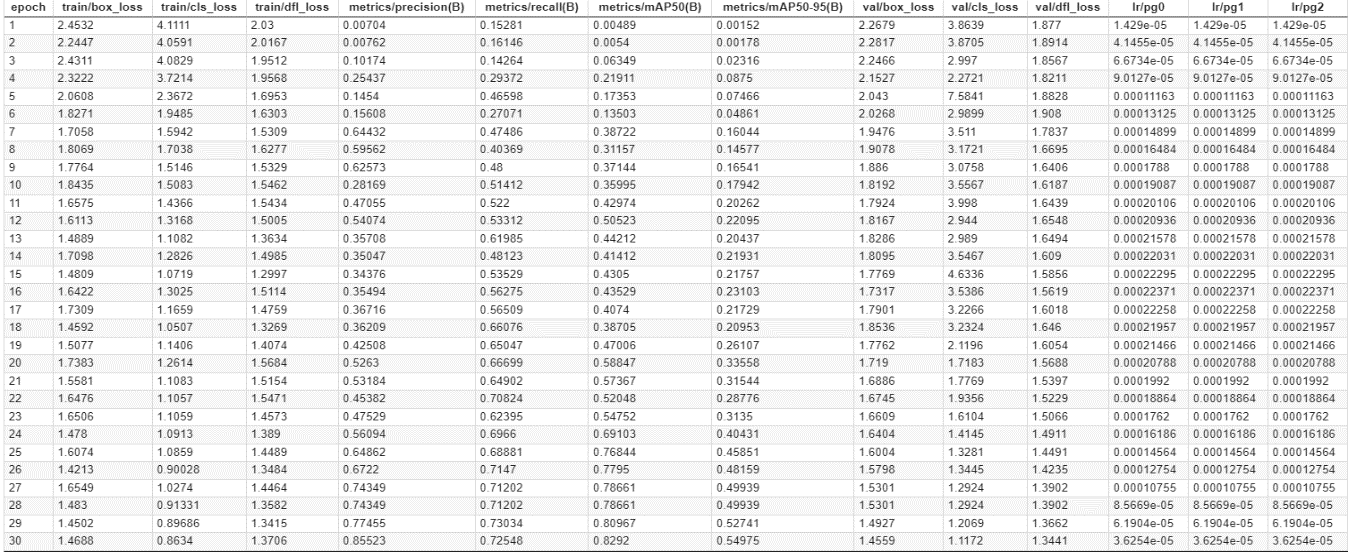


Figure : overview epochs

To make the table clear, we take a look at three columns, the training losses, the validation losses and the metrics.

Training losses: "train/box\_loss," "train/cls\_loss," and "train/dfl\_loss" decrease over the epochs. This is a positive sign, it indicates that the model is improving in terms of fitting the training data.

Validation losses: "val/box\_loss," "val/cls\_loss," and "val/dfl\_loss" also decrease over the epochs. Similar to the training losses, decreasing validation losses suggest that the model is performing well on data it hasn't seen during training.

Metrics increase: The metrics "metrics/precision(B)," "metrics/recall(B)," "metrics/mAP50(B)," and "metrics/mAP50-95(B)" are increasing over epochs, it suggests an improvement in the model's performance on the training set. A higher precision metrics indicates that the prediction positive instances are correct. A higher recall metrics indicates that the model is capturing a higher percentage of positive instances. Mean Average Precision at IoU (Intersection over Union) (mAP50(B)) threshold 0.5 is a metric commonly used in object detection tasks. An increasing mAP50 indicates that the model is becoming more accurate in locating and classifying objects. And last but not least, we have the metrics for mAP50-95(B). This metric considers mean average precision over a range of IoU thresholds from 0.5 to 0.95. An increasing mAP50-95 suggests improved performance across a broader range of IoU thresholds.

All in all, we could say that the model is improving itself over the time and indeed learning from itself.

## Of the results

In order to determine whether the expectations are in line with the result, we should see if we can give an answer to the question stated in the introductory paragraph: “make it clear for the software to see, based on a picture, whether there are empty shelves in a rack.”

I believe that, with confidence, I can say that the expectations are met. At least the foundation has been built. With this I mean that the model does it’s job, but it does have some flaws and shortcomings. In order to overcome those, more training data should be added to the model. In order to make the best out of the model, the camera that films the racks should be on a fixed place, this makes the task as easy as possible for the machine learning software and prevents the model from creating unnecessary mistakes. But overall, the model really did it’s job surprisingly well.

# Deployment

In order to deploy the AI model into the real life, some alterations and improvements still have to be made. The model did detect most of the empty shelves in the pictures I provided but did not do it entirely correctly when using new, more difficult pictures. This can be solved in two ways. Either dump a lot more pictures into the model and train it until it can’t go wrong anymore or standardize the way of visualising the data in order to make the task as easy as possible for the model.

I personally believe that, for the deployment of the model, the second option is the best option. When we look at the cost-benefit analysis, the first option would take much longer and it is even questionable whether it would improve the model or create so many alternative options that the model starts to fall apart.

The second option is better when it comes to cost-benefit and to the quality of the model. When the camera is at a fixed point, it is easier for the AI software to detect what part of the image is a product and what part is an empty shelve. This way, the model still works the same, but can detect changes in an easier way without the need of dumping hundreds or thousand of new pictures to finetune this.

Due to time related circumstances was I not able to directly implement the findings into the real life at ITEQ, which is a pity, but at least I can make them enthusiastic for the future.