Afbeelding met logo, symbool, Graphics, Lettertype

Automatisch gegenereerde beschrijving

Timon Thomassen (student)

15-12-2023

**Object detection  
POD detection**

**Title:**  Object detection

**Sub-title:** POD detection

**Student:** Timon Thomassen

2136595

E: [JT.Thomassen@student.han.nl](mailto:JT.Thomassen@student.han.nl)

**Study Background:** Logistics Engineering

**Class:** MDD-08 Data Science Tools and Techniques

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

 Ruitenberglaan 26

 6826 CC Arnhem   
 T: 024 353 0500

**Organization:** Jonker & Schut

**Lecturer:** John Smits

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

**Date:** 15-12-2023

**Place:** Arnhem

**Version:**  1.0

# Management Summary

This project aimed to enhance the efficiency of Jonker & Schut's photo validation process within their transportation department. Focused on Proof of Delivery (POD) photos, the current manual verification process is time-consuming for the planners due to poor photo quality. The solution involved implementing a YOLOV8 model for photo classification, providing a quick and accurate assessment of photo quality.

The YOLOV8 model, trained on a carefully curated database, effectively categorized photos into distinct classes such as blurry, incomplete, skewed, and good. Through data preparation and annotation using Roboflow, the model exhibited promising results after 30 epochs. This demonstrated its ability to distinguish between acceptable and poor photos.

To deploy this solution within Jonker & Schut's operations, a streamlined integration process was proposed. The YOLOV8 model would be incorporated into the existing pipeline, where incoming photos encoded in BASE64 format would be routed through the model for quick assessment. Subsequently, the model would generate a list of transport orders with problematic photos, aiding planners in identifying and rectifying errors swiftly.

The proposed integration also leveraged the ERP system's capabilities, introducing "indication stacks" to filter and present a list of orders with faulty photos directly to the planners. This user-friendly approach aimed to significantly reduce the time required for manual checks.

In conclusion, the YOLOV8 model, when seamlessly integrated into Jonker & Schut's workflow, successfully met the project's objectives. It accurately classified photos, generated a list of orders with photo issues, and streamlined the manual verification process for planners. This holistic solution aligns with Jonker & Schut's commitment to quality and service, showcasing the potential for innovative technologies to optimize logistics operations.

Table of Contents

[Chapter 1: Business understanding 4](#_Toc153542208)

[1.1 General overview Jonker & Schut 4](#_Toc153542209)

[1.2 POD photos 4](#_Toc153542210)

[1.3 Goal of the research 4](#_Toc153542211)

[Chapter 2: Data understanding 5](#_Toc153542212)

[Chapter 3: Data preparation 6](#_Toc153542213)

[3.1 Jonker & Schut database 6](#_Toc153542214)

[3.2 Database for YOLOV8 model 6](#_Toc153542215)

[3.3 Photo annotation (Roboflow) 7](#_Toc153542216)

[Chapter 4: Modelling 8](#_Toc153542217)

[4.1 YOLOV8 explanation 8](#_Toc153542218)

[4.2 Python script 8](#_Toc153542219)

[Chapter 5: Evaluation 10](#_Toc153542220)

[chapter 6: Deployment 11](#_Toc153542221)

[6.1 Integration into Jonker & Schut 11](#_Toc153542222)

[6.2 Conclusion 13](#_Toc153542223)

[Chapter 7: Appendix 14](#_Toc153542224)

[7.1 Appendix A: wrong POD photos 14](#_Toc153542225)

[7.2 Appendix B: Good POD photo 19](#_Toc153542226)

[7.3 Appendix C: Python script 20](#_Toc153542227)

[7.4 Appendix D: Github link 21](#_Toc153542228)

# Chapter 1: Business understanding

## 1.1 General overview Jonker & Schut

This project is conducted on behalf of Jonker & Schut, also known as J&S. J&S has fifteen different locations spread across Barneveld, Amersfoort, Putten, and Nijkerk. Jonker & Schut is a logistics specialist in the food industry, with a focus on powdered dairy products. They also excel in the storage and handling of organic goods.

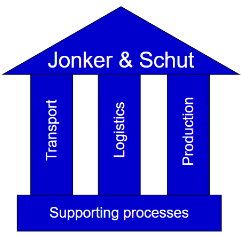


Figure 1: Jonker & Schut

See figure 1 for the main departments of J&S. Transportation planning is carried out by the transportation department. The production department handles various production lines, and logistics is responsible for internal warehousing. J&S also has its own customs department for customs clearance.

Jonker & Schut's unique selling point is a strong emphasis on both quality and service, with a slight emphasis on quality. In total, J&S employs about 200 staff members. The transportation department, where this internship takes place, consists of six individuals in the office and approximately 40 drivers. Jonker & Schut has 35 trucks and 90 trailers, along with four terminal tractors that regulate the flow between the various J&S locations (Thomassen, 2023).

The research is done within the transportation department of Jonker & Schut.

## 1.2 POD photos

The transportation department of Jonker & Schut transports goods all around Europe. When the goods are delivered the customer must sign a POD, a Proof Of Delivery. With this signed POD Jonker & Schut can proof that they delivered the goods (Creative logistics solutions, N.D.). Jonker & Schut uses a board computer to instruct their drives on the shipments they must deliver. When they deliver the goods at the destination the customer must sign the POD, after that the driver takes a photo of the POD via the board computer. Because the board computer is linked to the TMS (Transport Management System) the photo is automatically placed behind the correct transport order. But right know the quality of those photo’s isn’t good enough. A lot of the photos are not good enough, blurry or only half, to serve as a proof of the delivery.

Because of this the planners at Jonker & Schut check all the Transport Orders manually to see for which order the photos are wrong. Those photos are replaced by a scan made at the office by the planner. This is a very time consuming process.

## 1.3 Goal of the research

All the POD’s photos are taken and these will be checked for quality. To ensure a good result the following objective is stated:

1. The model should be able to classify new photos correctly as a good or wrong photo. The chosen model is a YOLOV8 model.
2. Create a list of the wrong photos to provide the planners of Jonker & Schut with a list of orders were the photos have to be redone.
3. Make sure the process of checking the photos done by the planners takes less time.

# Chapter 2: Data understanding

The data that is used in the research contains of POD photos that were taken by the drivers of Jonker & Schut. There are 6 different categories of photos:

1. Blurry: The information on the POD isn’t readable. These photos are wrong.
2. Incomplete: The photo doesn’t show the entire POD, this means that not all information is available. These photos are wrong.
3. Skewed: The photo is turned 90 degrees and cropped into that format. This means that the photo doesn’t look professional to the customer and is sometimes not readable. These photos are wrong.
4. Too much shown: The photo shows more that only the POD, for example the steering wheel of the truck. This means that the photo doesn’t look professional to the customer and is sometimes not readable. These photos are wrong.
5. Hand of driver: The hand of the driver can be seen on the photo. This means that the photo doesn’t look professional to the customer and is sometimes not readable. These photos are wrong.
6. Good: These photos are readable, show all the information, aren’t skewed and don’t contain more than the POD or the hand of the driver. These photos are good.

In Appendix A and B examples of the photos can be found. In appendix a examples of photos 1 till 5 and in appendix 6 an example of photo 6.

The initial database of Jonker & Schut contains all the photos from all the orders that are done until this day. From this big database photos were selected for the database that was used in the model, see the next chapter.

# Chapter 3: Data preparation

To prepare the data for analysis the photos of the POD’s are checked. Is the photo good enough for the database? After that the right photos are annotated using Roboflow.

## 3.1 Jonker & Schut database

A lot of different transport orders were gone through to make sure the right photos are collected. The photos that are wrong should be very wrong, photos that are doubtable wrong are not put in the database. The same goes for the good photos. This makes sure that the model has a good database to base the photo classification on.

Because a lot of photos that are wrong are deleted and manually replaced with good photos the historical database of Jonker & Schut didn’t provided enough photos that are wrong. To make up for the lack of data own photos were taken.

## 3.2 Database for YOLOV8 model

After the above described selection of data the following database was put together. Based on that database the object detection model is trained.

|  |  |
| --- | --- |
| **Category** | **Number of photos** |
| Blurry | 11 |
| Incomplete | 5 |
| Skewed | 2 |
| Too much shown | 4 |
| Hand of driver | 3 |
| Good | 36 |

In appendix A multiple wrong photos can be found. There are a few categories of wrong photos.

* A blurry photo
* An incomplete photo
* A skewed photo
* A photo which shows too much
* A photo with the hand of the driver on it

In appendix B a photo which is good can be found.

## 3.3 Photo annotation (Roboflow)

By the use of Roboflow the photos were annotated, that means that a certain class was added to the photo. Based on these classes the python script can detect if a new photo also belong to one of the classes.   
In figure 2 a wrong photo which has two classes can be found. The yellow class is the annotation “Wrong – too much” and the light blue class is annotated as “Wrong – Hand”. In figure 3 a good photo can be found. The dark blue class is annotated as “Good”.



Figure 2: Annotated photo (2)

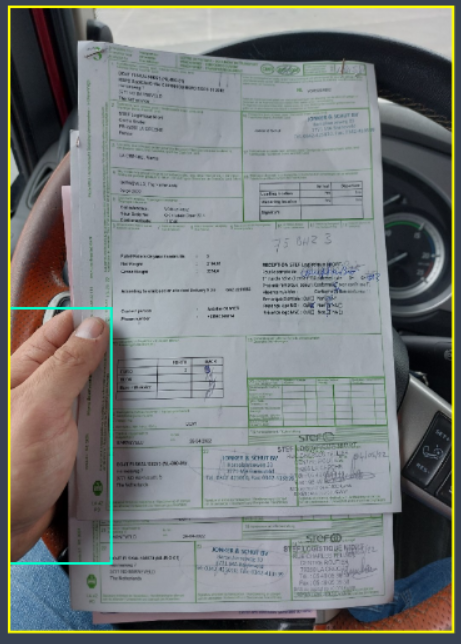


Figure 3: Annotated photo (1)

Figure 4 shows the collection of all the photo’s in the app roboflow, this can also be found in appendix C

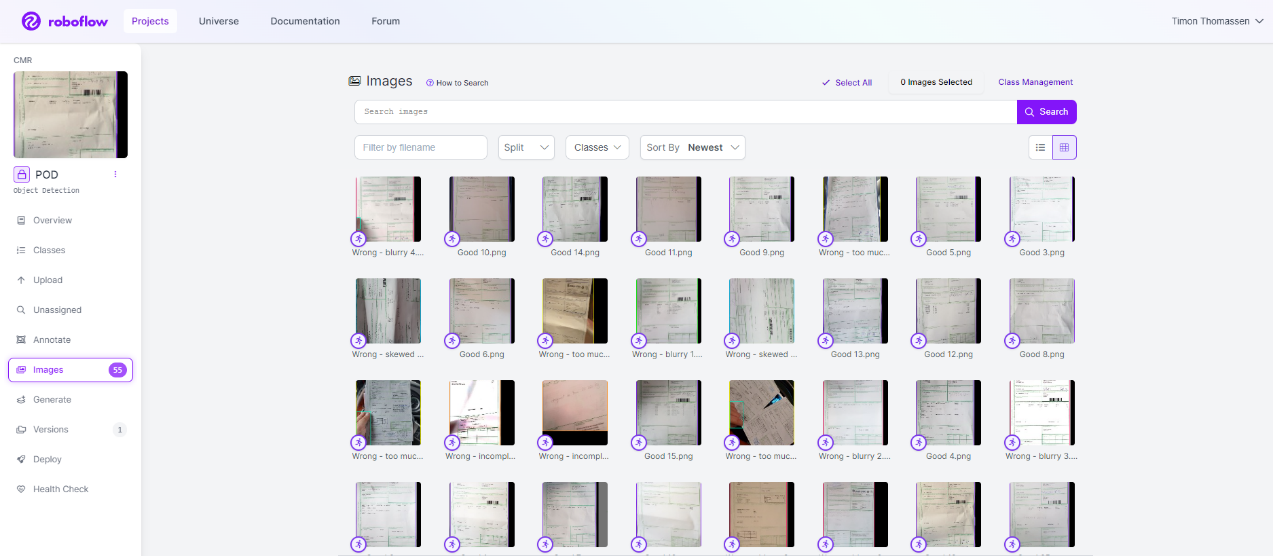


Figure 4: Photos in Roboflow

# Chapter 4: Modelling

A python script is conducted to do the detection of new images. This is done via the YOLOV8 model.

## 4.1 YOLOV8 explanation

YOLOV8 is the newest YOLO model that can be used for object detection, image classification and instance segmentation. This means that the model can recognize an object or classify a image based on the database it has been given.

In figure 5 an example for YOLOV8 object detection can be found. The model recognized two persons with 81 and 79% accuracy and a tie with 37% accuracy. The same is done in this assessment, based on a database of photos the YOLOV8 model is tasked with recognizing if a new photo is wrong or good.

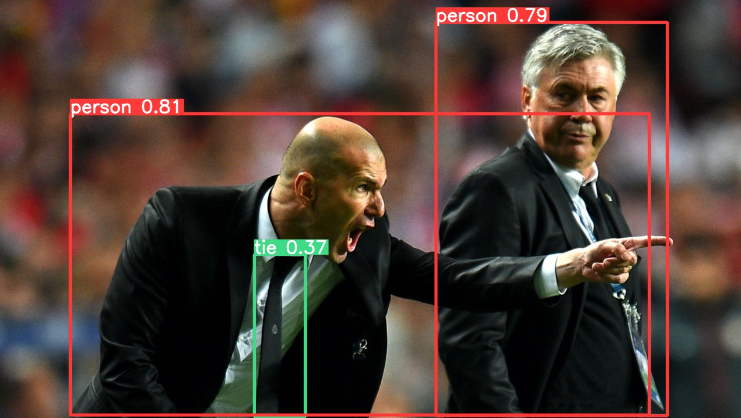


Figure 5: YOLOV8 example

## 4.2 Python script

In appendix C the python script of the YOLOV8 model can be found. When the python script was run with 55 photos in the database the following error message occurred, see figure 6 Based on the ChatGPT information of this error message I couldn’t find a way to fix it. But the prediction that was done, see next chapter, succeeded anyway. So at this moment nothing is done about this. If the error message has a impact in the future, actions will be taken accordingly.



Figure 6: Error message python script

To make sure that the model is accurate in its prediction a certain amount of epochs are done. A epoch represents each time a dataset passes through the algorithm. The higher the amount of epochs, the better the accuracy of the model will be. Each time the dataset passes through the algorithm, the weights of the algorithm will be updated to optimize the learning (Simply learn, 2023).

In this model 30 epochs are used. At first the model was run with 5 epochs, but the algorithm classified al the test photo’s as good. This can be found in the green box in figure 7.

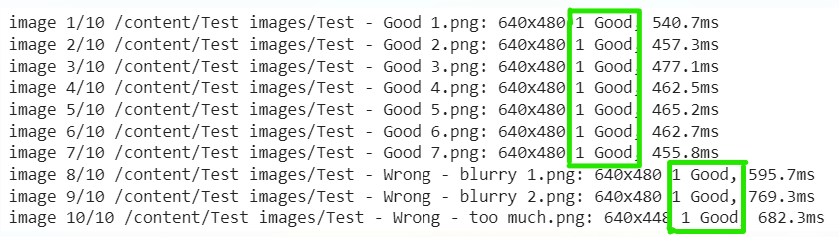


Figure 7: Test run with 5 epochs

This meant that the number of epochs was way to low. After that the model was tried with 30 epochs, this made the predictions accurate.

# Chapter 5: Evaluation

To evaluate the model a test run is done with the following photos:

* 7 good photos
* 2 blurry photos
* 1 photo which shows too much

The model was fairly accurate in its prediction. It classified all the photo’s correct, see the green boxes in figure 8.



Figure 8: Results model

But for image 8, which was wrong, the model also classified the photos as “good”, the model gave two classifications. This could be dealt with in two ways:

1. Add more blurry photos to the dataset to make accurate predictions
2. Keep the model this way. If the creation of the list with photos that are wrong contains photos with all wrong classification this photo would still be on that list.

This means that the first objective that was stated in chapter 1 is met. Objective 2 and 3 are not met in this research already. But there are future plans to integrate this model in the process of Jonker & Schut, then those objectives will also be met. See chapter 6 for the design process for the deployment of this integration.

# chapter 6: Deployment

## 6.1 Integration into Jonker & Schut

Right now the model doesn’t produce a list with orders that have a wrong photo. But there are future plans to integrate this model in the pipeline of Jonker & Schut. This isn’t done during this research but will be done as part of my Job at Jonker & Schut.  
The following process is designed, see figure 9.

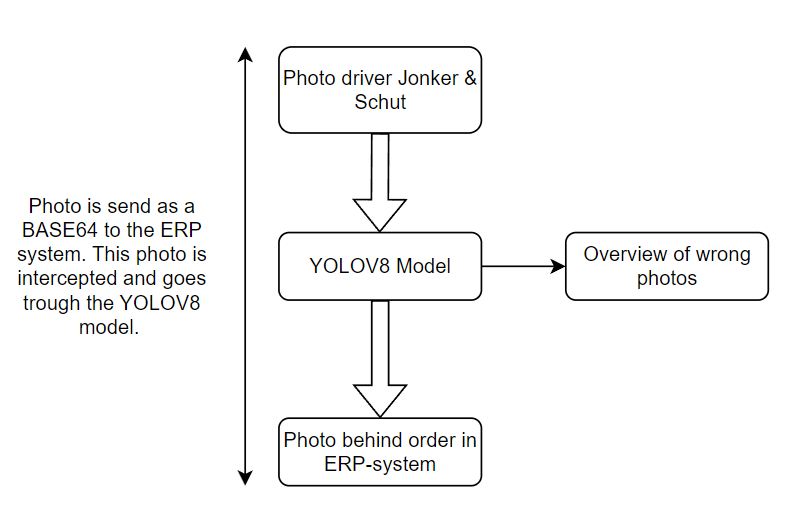


Figure 9: Integration YOLOV8 into process Jonker & Schut

The integrate the YOLOV8 model into the process of Jonker & Schut the following steps should be taken:

* BASE64 intercepted and pushed through the YOLOV8 model.
* BASE64 pushed in ERP system as it is done in the current process
* The YOLOV8 model should make an overview of the wrong photos

**BASE64 intercepted and pushed through the YOLOV8 model**The photos is via an message which contains a BASE64 file of the photo. This photos should be intercepted and pushed through the model. This step can be done by using the current pipeline of Jonker & Schut. The Senior Application Administrator of Jonker & Schut is going to take care of this step.

**BASE64 pushed in ERP system as it is done in the current process**After the BASE64 file is pushed through the YOLOV8 model, the photo has to be send to the correct transport order in the TMS. This step can be done by using the current pipeline of Jonker & Schut. The Senior Application Administrator of Jonker & Schut is going to take care of this step.

**The YOLOV8 model should make an overview of the wrong photos**In the message were the BASE64 file is send to the system the name of the Transport Order is mentioned, see figure 10. Based on this name the YOLOV8 model will create a SQL table with all the Transport Orders were the photo is wrong.



Figure 10: BASE64

This SQL table is linked to the ERP system that Jonker & Schut uses. Inside the ERP system, it is possible to create so called “indication stacks”. These are buttons that will put a predetermined filter on a chosen database. These indication stacks will be used to provide a list of Transport Orders which have wrong photos to the planner. The database that is going to be used is “TMS document header” and the filters are as follows:

* Documentsoort: “Order”
* POD ontvangen: “ja”
* POD fout: “ja”

Based on this indication stack the transport planner can very easily correct the orders with wrong photos.

## 6.2 Conclusion

After this integration into the pipeline of Jonker & Schut all three objectives as stated in chapter 1.3 are met.  
The objectives were as follows:

1. The model should be able to classify new photos correctly as a good or wrong photo. The chosen model is a YOLOV8 model.
2. Create a list of the wrong photos to provide the planners of Jonker & Schut with a list of orders were the photos have to be redone.
3. Make sure the process of checking the photos done by the planners takes less time.

**Objective 1:**   
This is met by the use of the YOLOV8 model that is trained by the database created at Roboflow

**Objective 2:**This is met by the use of the BASE64 name. In this name the Transport Order is mentioned. And based on that name a list with orders who have a wrong photos is made.

**Objective 3:**This is met by the use of the so called “indication stacks” in the TMS of Jonker & Schut. With one click the transport planner has a list of orders who have a wrong photo.

# Chapter 7: Appendix

## 7.1 Appendix A: wrong POD photos

**Blurry photo**

**Afbeelding met tekst, document, Parallel, ontvangst

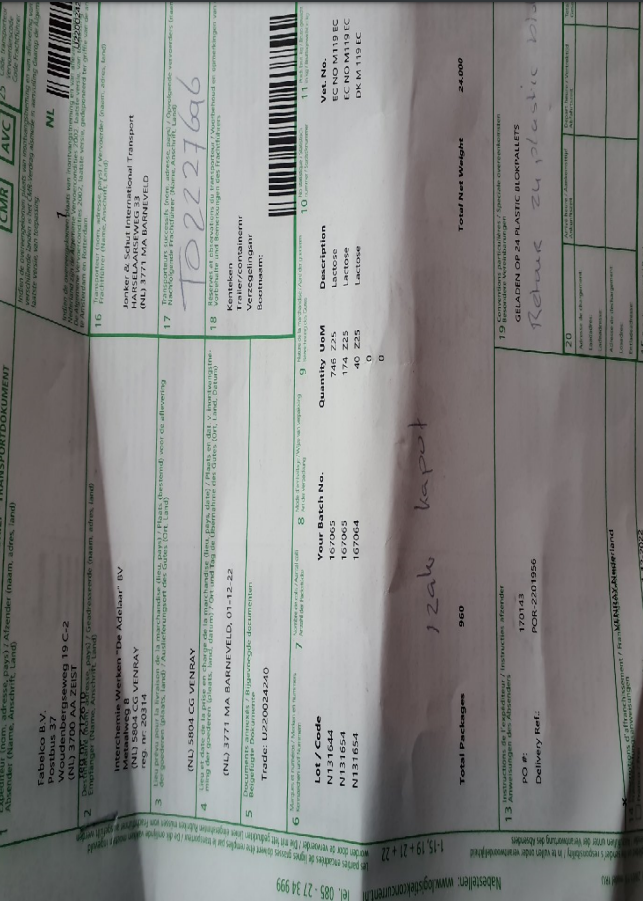
Automatisch gegenereerde beschrijving**

**Incomplete photo**

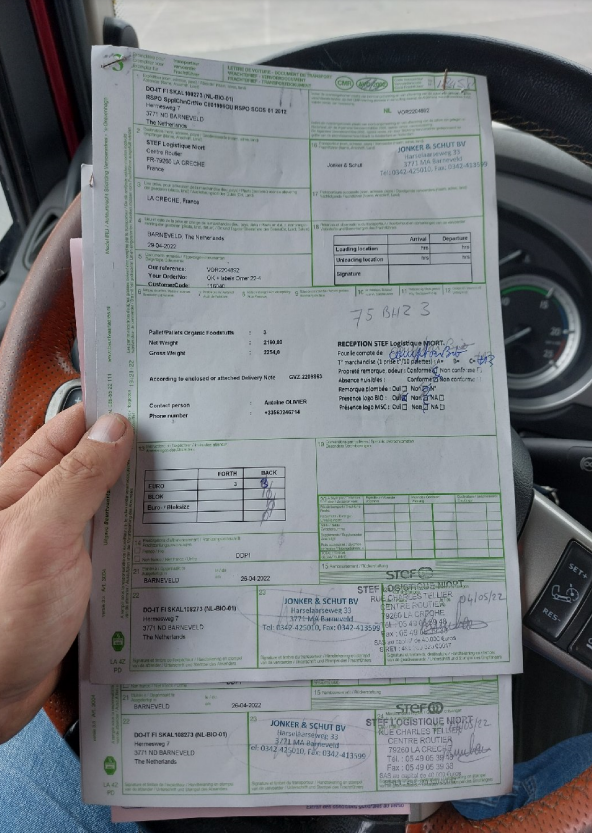
**Afbeelding met tekst, handschrift, schermopname, Parallel

Automatisch gegenereerde beschrijving**

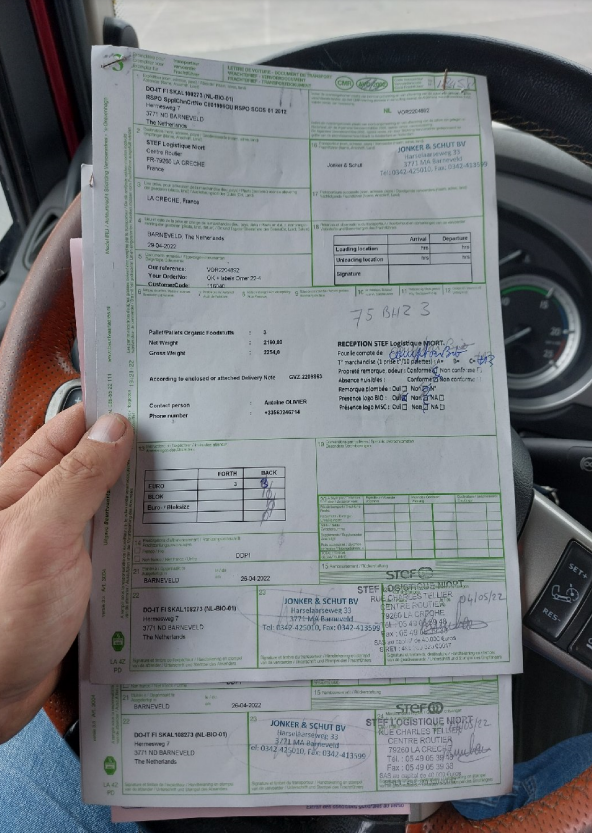
**Skewed photo**

****

**Photo which shows too much**

****

**Photo with hand of driver on it**

****

## 7.2 Appendix B: Good POD photo



## 7.3 Appendix C: Python script

**File**



**Screenshots**

**Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving**

**Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving**

**Afbeelding met tekst, Lettertype, lijn, schermopname

Automatisch gegenereerde beschrijving**

**Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving**

**Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving**

## 7.4 Appendix D: Github link

<https://github.com/TimonThomassen/POD-recognition>