AGILEHAND

D4.2 – AGILEHAND Smart Sensing SUITE v2

WP4 – BUILD: AGILEHAND Smart Sensing SUITE

AGILEHAND has received the funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101092043.

AGILEHAND

Document Information

GRANT AGREEMENT NUMBER	101092043	ACRONYM	AGILEHAND
FULL TITLE	Smart grading, handling, and packaging solutions for soft and deformable products in agile and reconfigurable lines		
START DATE	01-01-2023	DURATION	36 months
PROJECT URL	https://agilehand.eu/		
DELIVERABLE	D4.2 – AGILEHAND Smart Sensing SUITE v2		
WORK PACKAGE	WP4 – BUILD: AGILEHAND Smart Sensing SUITE		
DATE OF DELIVERY	CONTRACTUAL: December 2024	ACTUAL	-
NATURE	Other	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	FBK		
RESPONSIBLE AUTHOR	Dr. Mohamed Lamine Mekhalfi		
CONTRIBUTIONS FROM	FBK, UNINOVA, UPM		
TARGET AUDIENCE	1) AGILEHAND Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community		
DELIVERABLE CONTEXT/ DEPENDENCIES	This document is the second iteration of WP4. Its relationship to other documents is as follows: - D4.1 Title: AGILEHAND Smart Sensing SUITE v1		
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None		
READING NOTES	None		
ABSTRACT	The scope of D4.2 is to showcase the final progress developed within WP4 which regards the development of Smart Sensing SUITE. In this respect, smart sensing, data acquisition, quality grading and self-calibrating aspects of the proposed solutions are reported and discussed.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	11/11/2024	ToC and working version	Created ToC and document structure	FBK, UNI
0.2	28/11/2024	Working version	AGILEHAND Data-Sets and Grade the Quality sections draft	FBK
0.3	29/11/2024	Working version	AGILEHAND Grade the Quality section draft rectification	FBK
0.4	05/12/2024	Working version	AGILEHAND Self-Calibrating section draft	UNI
0.5	07/12/2024	Working version	Formatting of the document	UNI
0.6	10/12/2024	Final	Final version for internal review	FBK
1.0	20/12/2024	Final	Internal reviews addressed	FBK

Disclaimer

Any dissemination of results reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
CR^{Hand}	Collaborative Robot Handling
DS^{Sense}	Data Sets Sensing
DT^{Agile}	Data Driven Digital Twin
ETA	Estimated time of arrival
GQ^{sense}	Grade the Quality Sensing
IoT	Internet of Things
ICT	Information and Communication Technologies
KER	Key Exploitable Result
KPI	Key Performance Indicator
MES	Manufacturing Execution System
ML	Machine Learning
OER	Other Exploitable Result
OPC/UA	Open Platform Communications/Unified Architecture
PE^{Agile}	Production Execution Optimization Toolkit
PLC	Programmable Logic Controller
PR^{Agile}	Production Reconfiguration
PT^{Agile}	Product Oriented Traceability
RR^{Hand}	Robot Robot Handling
SC^{sense}	Self-Calibrating Sensing
SMED	Single Minute Exchange of Dies
ST^{hand}	Self-Adaptable Transportation System
UI	User Interface
WP	Work Package

Executive summary

This deliverable provides an update regarding the activities carried out within WP4 with **AGILEHAND** pilots, namely Multiscan, Sant'Orsola, Produmar and Marelec, with respect to the reported detailed in Deliverable D4.1. In particular, self-calibration, quality grading, dataset acquisition, and smart sensing solutions are described in the following sections.

Document structure

Section 1 describes the updates to the work carried out on the "AgileHand Self-Calibrating" solution, developed in Task 4.1, describing the solution overview and the implementation status of version 2.0.

Section 2 describes the updates to the work carried out on the "AgileHand Grade the Quality" solution, developed in task 4.2, describing the solution overview and the implementation status of version 2.0.

Section 3 describes the updates to the work carried out on the "AgileHand Data-Sets" solution, developed in task 4.3, describing the solution overview and the implementation status of version 2.0.

Section 4 updates the status of the functional and implementation viewpoints v2 of "AgileHand Smart Sensing" suite.

Section 5 includes the conclusions of the document and of the work carried out in the three tasks of WP4.

1. AGILEHAND Self-Calibrating

1.1. Overview

This section continues the reporting of the **AGILEHAND Self-Calibrating** solution whose details were presented in D4.1. This section describes the continued progress regarding the deployment of the sensors that were previously identified and are used for each pilot.

1.2. Status

The hardware requirements were surveyed and the set of sensors that build the Smart Sensing System have been defined and tested in all 4 pilots. Support for the traceability system is under development with the use of a sensing microcontroller board design that is capable of sustaining the harsh operating conditions of the different use cases.

1.2.1. Release v2.0

1.2.1.1. Multiscan

The sensor setup for **Multiscan** pilot has already been detailed in Deliverable D4.1 and has not been changed or updated.

1.2.1.2. Sant'Orsola

It is to recall that for this pilot, the goal is to grade raspberries inside a punnet already in motion along a conveyor belt, which is a challenging but worth-addressing task due to the small size of the fruits on the one hand, and their occlusion on the other (i.e., a punnet contains at least two layers of fruits). Three sensing solutions were explored over the course of the first release for this pilot. RealSense™ D415, RealSense™ D456 and a stereo vision system (LIR Blackfly BFLY-PGE-13E4C-CS) were considered.

The first option, RealSense™ D415, did not provide optimal visibility due to its rolling shutter, which produces a blurring effect. Similarly, the stereo vision system suffered from the blurring effect, besides the fact that it captures poorly illuminated images. The third sensor, RealSense™ D456, performed the best. However, as explained in the previous release, It is important to leverage only the RGB images but also the respective depth data which is essential to localise the raspberries inside the punnet in 3D world and provide this information to the robotic system (envisioned in WP5) for further pick and place operations. On this point, this sensor must be place at a minimum range (60 cm) to enable depth acquisition, which comes at the cost of low resolution RGB images. It was concluded that the RGB and depth data may be acquired separately.

Therefore, another data collection was made by placing the RealSense™ D456 sensor much closer to the punnets for the RGB acquisition only. The sensor was placed at 12 cm from the conveyor belt. This distance was determined based on trials at the test site, and the camera parameters were tuned until the best visibility was obtained. The camera was fixed on a tripod and attached to a laptop via a USB cable. The acquisition equipment and setup from different viewpoints in illustrated in **Figure 1.**

Figure 1: Acquisition setting for Sant'Orsola use case.

Moreover, lighting conditions were drastically improved with respect to the previous release by adopting a 1000 lumen torch from Wuerth (**Figure 2** top) placed at 25 cm from the conveyor plane, which was fixed to a medallic pole via its built-in magnet (**Figure 2** bottom) facing the punnet of interest. For instance, the different in illumination between the punnet of interest (the one right below the camera) exposed to the afore-mentioned torch and the punnets lining up on the conveyor belt that are exposed to ambient lighting can be observed at the bottom examples of **Figure 2**.

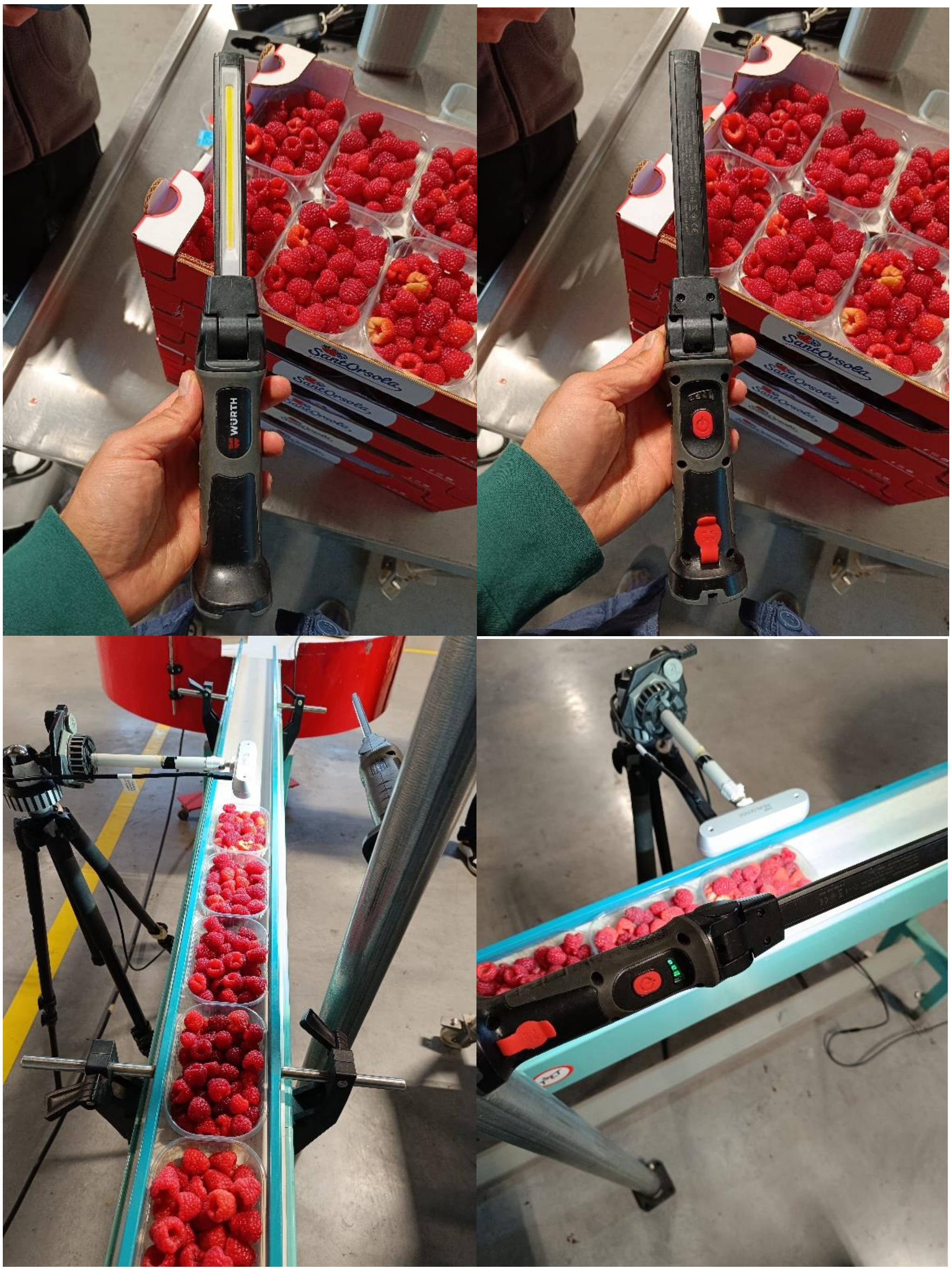


Figure 2: Illustration of the illumination torch and the difference in illumination between the target punnet and the remaining ones.

In this respect, to better evidence the difference in image quality between the old and the new acquisition setups is depicted in the examples of **Figure 3** top and bottom rows respectively**.** For instance, the previous setting allows the perception of the overall punnet and fruit distribution while it does not portray fine fruit-level details that are essential to enable colour-based grading. The new setting, however, features the details of each raspberry regarding both the colour and the texture. This was also confirmed with the experts of **Sant'Orsola**. It is to mention that in the previous setup, a resolution of 1280x720 was opted for to enable the RGB and depth frames (i.e., the depth frames can be acquired at a maximum resolution of 1280x720), while this time the resolution is upgraded slightly to 1280x800 as only RGB images were acquired, which helps gain more visible details. The frame rate is 30fps.

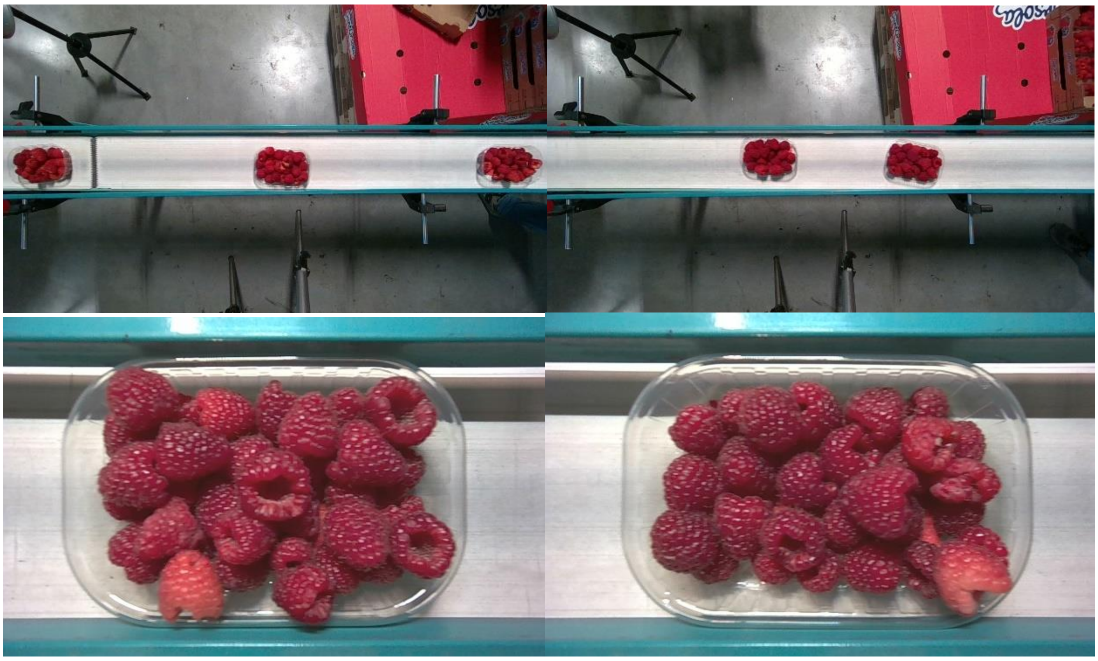


Figure 3: Difference in resolution and fruit-level details between the first (top row) and the second (bottom images) releases. Images captured with the same sensor (RealSense™ D456).

1.2.1.3. Produmar

The setup used for image capturing in **Produmar** has already been detailed in Deliverable D4.1 and it was not necessary to be updated.

Regarding the support for the traceability system, the sensor pack is under development with the use of a temperature sensing microcontroller board, designed to sustain the harsh operating conditions of the very cold and humid production line in **Produmar**. This development will continue to be carried out in articulation with **Task T6.1** and its progress detailed in **Deliverables D6.1** and **D6.2**.

1.2.1.4. Marelec

A similar acquisition protocols as in **Produmar** is adopted. In particular, a RealSense™ D456 camera was mounted on top of a tripod at about 57 cm from the conveyor plane. The distance of the camera from the target fillets as well as its parameters were tuned on site until satisfactory images were obtained, then the acquisition took place. The camera was connected to a laptop via a USB cable. Above the conveyor, there was a V-shaped enclosure that contained LED lightings. The entire acquisition setup and its equipment are given in **Figure 4**.

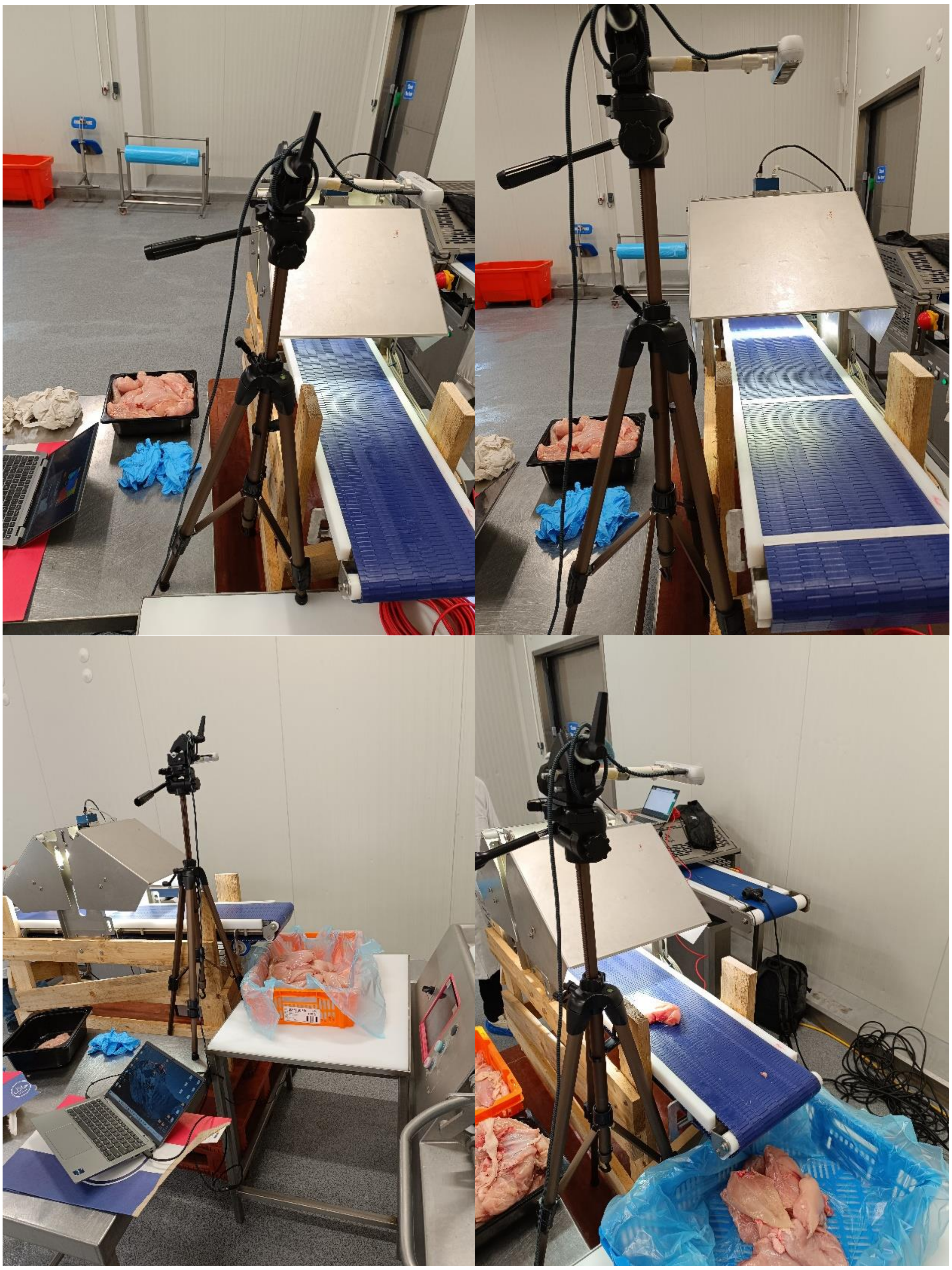


Figure 4: Acquisition setup for Marelec pilot.

The images were acquired at a frame rate of 30 fps and resolution of 1280x720. However, to focus the process on the conveyor belt area of interest, the images were cropped to frames of size 700x300 and discarded the other parts as they contain chicken trays that may pose bottlenecks in the grading task (e.g., false positive detections outside the conveyor region).

2. AGILEHAND Grade the Quality

2.1. Overview

The grading part of AGILEHAND incorporates artificial intelligence (AI) data-driven methodologies owing to their cutting-edge performance [1]. For all the use cases envisioned in the project, a step of instance segmentation is essential to locate the different food products in the scene of interest. Therefore, the precision of the earlier step is paramount as it affects later steps such as 3D localisation, which serves other subsequent operations (i.e., robotic product manipulation considered in WP5). The first release focused more on the acquisition of qualitative and quantitative data that enables reliable grading scores. In what follows, details are provided regarding the grading outcomes of the use cases that were not addressed in the previous release (before delving into this section, it is recommended to first read the dataset acquisition section since technically it comes first prior to grading). Moreover, a user interface was developed to enable the visualisation of the grading outcomes for each pilot.

2.2. Status

2.2.1. Release v2.0

2.2.1.1. Multiscan

For this pilot, the grading task was already concluded in the first release, where fruit-level grading of oranges into three classes was tackled. Subsequently, it was decided upon discussions with **Multiscan** to expand the grading process to a crate-level grading (**Figure 5**). In particular, the aim is to grade the top layer of a whole crate and obtain a rough histogram representation according to several criteria such as shape, size, and colour. The grading histograms will be leveraged to fine tune the parameters of the fruit-level grading machines at **Multiscan**.

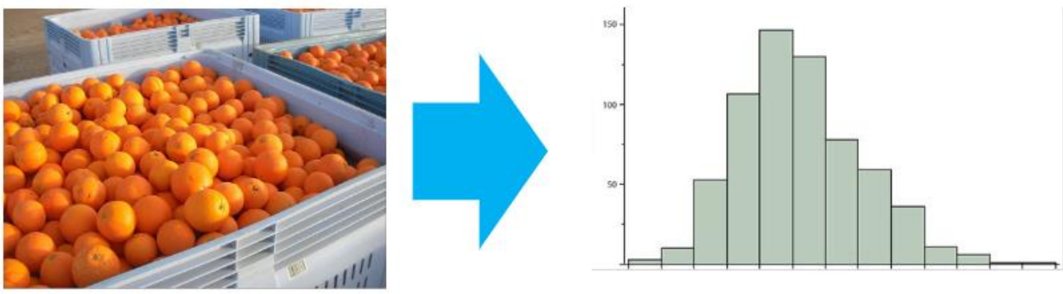


Figure 5: Overall crate-level orange grading task.

Currently, the plan is to conduct a dataset acquisition that enables crate-level analysis of oranges. In view of the grading pipeline, the algorithms were already set up in place. First, once the dataset was acquired and annotated, instance segmentation is performed to isolate single fruits inside the crate (**Figure 6**). Second, quality measurements of each fruit are conducted. Third, the latter measurements are combined to form a histogram pertaining to the crate of interest. Regarding the instance segmentation step, YOLOv11 [2] will be trained on the annotated data on account of its speed and precision.

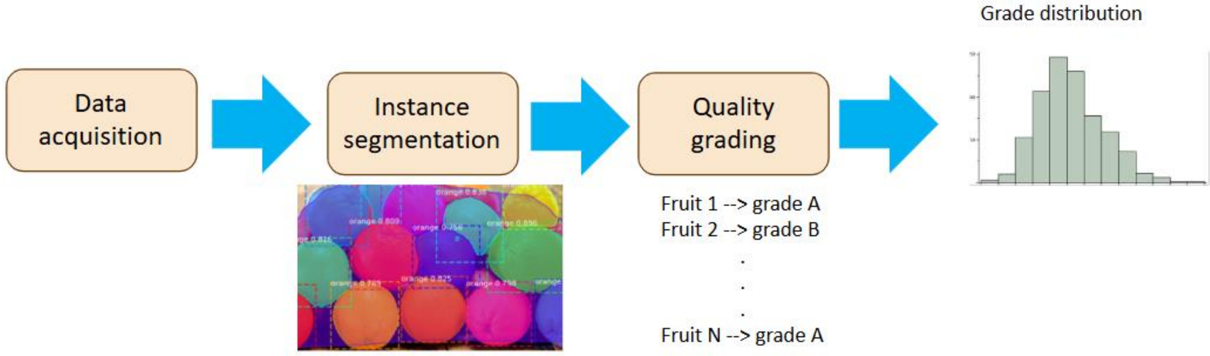
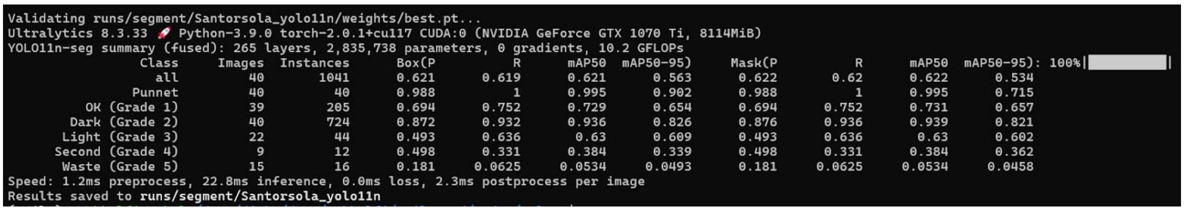


Figure 6: Crate-level orange grading pipeline.

2.2.1.2. *Sant'Orsola*

The main challenge that characterised the requirements of the **Sant'Orsola** use case was the resolution of the images in the first release. In this release, this bottleneck was addressed successfully by improving the image quality drastically. Therefore, YOLOv11 model was trained for 100 Epochs based on the pretrained yolo11n-seg backbone on the training data and validated it on the validation data. The implementation was developed in Python, and the grading scores are reported in the screenshot **Figure 7** in a range of 0 to 1 (1 being the best i.e. 100%). For instance, the screenshot reports bounding box and mask scores, which refer to the predicted bounding box surrounding a target object (punnet or raspberry) and the mask of all the predicted pixels that belong to each target object, respectively.

The P, R and mAP50 and mAP50-95 are common metrics in object detection and instance segmentation. Precision (P) measures the accuracy of positive punnet/raspberry detections. The Recall (R) measures the ability of the model to detect all relevant punnets/raspberries. mAP50 is the mean Average Precision at an Intersection over Union (IoU) threshold of 0.5. It evaluates the average precision across all punnet/raspberry classes at a fixed IoU threshold of 0.5. It is adopted to determine how well the model detects objects correctly (both in terms of precision and recall). mAP50-95 is a stringent version of mAP50, where the mAP is computed at multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05 [3-4].



```
Validating runs/segment/Santorsola_yolo11n/weights/best.pt...
Ultralytics 8.3.33 🚀 Python-3.9.0 torch-2.0.1+cu117 CUDA:0 (NVIDIA GeForce GTX 1070 Ti, 8114MiB)
YOLO11n-seg summary (fused): 265 layers, 2,835,738 parameters, 0 gradients, 10.2 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|
                   all         40       1041      0.621      0.619      0.621      0.563      0.622       0.62      0.622      0.534
                Punnet         40         40      0.988          1      0.995      0.902      0.988          1      0.995      0.715
           OK (Grade 1)        39        205      0.694      0.752      0.729      0.654      0.694      0.752      0.731      0.657
         Dark (Grade 2)        40        724      0.872      0.932      0.936      0.826      0.876      0.936      0.939      0.821
        Light (Grade 3)        22         44      0.493      0.636       0.63      0.609      0.493      0.636       0.63      0.602
       Second (Grade 4)         9         12      0.498      0.331      0.384      0.339      0.498      0.331      0.384      0.362
        Waste (Grade 5)        15         16      0.181     0.0625     0.0534     0.0493      0.181     0.0625     0.0534     0.0458
Speed: 1.2ms preprocess, 22.8ms inference, 0.0ms loss, 2.3ms postprocess per image
Results saved to runs/segment/Santorsola_yolo11n
```

Figure 7: Punnet and raspberry grading scores on the Sant'Orsola dataset.

In the results of **Figure 7**, the punnets were detected at a mAP50 of 0.995, which is evident as they are large distinguishable objects. Regarding the five classes of raspberries, Grade 2 was the easiest to detect at a mAP50 of 0.934 due to the high number of training samples in this class, followed by Grade 1, Grade 3 and Grade 4 which have less training examples in the dataset compared to Grade 2. The most difficult to detect was Grade 5, owing to the rather small number of training instances in this class. In fact, Grade 5 represents the 'waste' class and should be detected and localised properly by the grading system to enable a correct handling by the robotic solution (WP5). We believe that the extension of the current dataset with more samples from Grades 3, 4, and 5 is ought to balance out the representation of the different fruit classes and entail higher grading scores for each. In this regard, another supplementary acquisition session is currently being planned with **Sant'Orsola**. Afterwards, the already trained YOLOv11 model will be finetuned on the new data. A qualitative grading example is given in **Figure 8** (note that the numbers at the top of each bounding box detection refer to the grading confidence in a range of 0 to 1, this also applies in following examples regarding the other use cases).

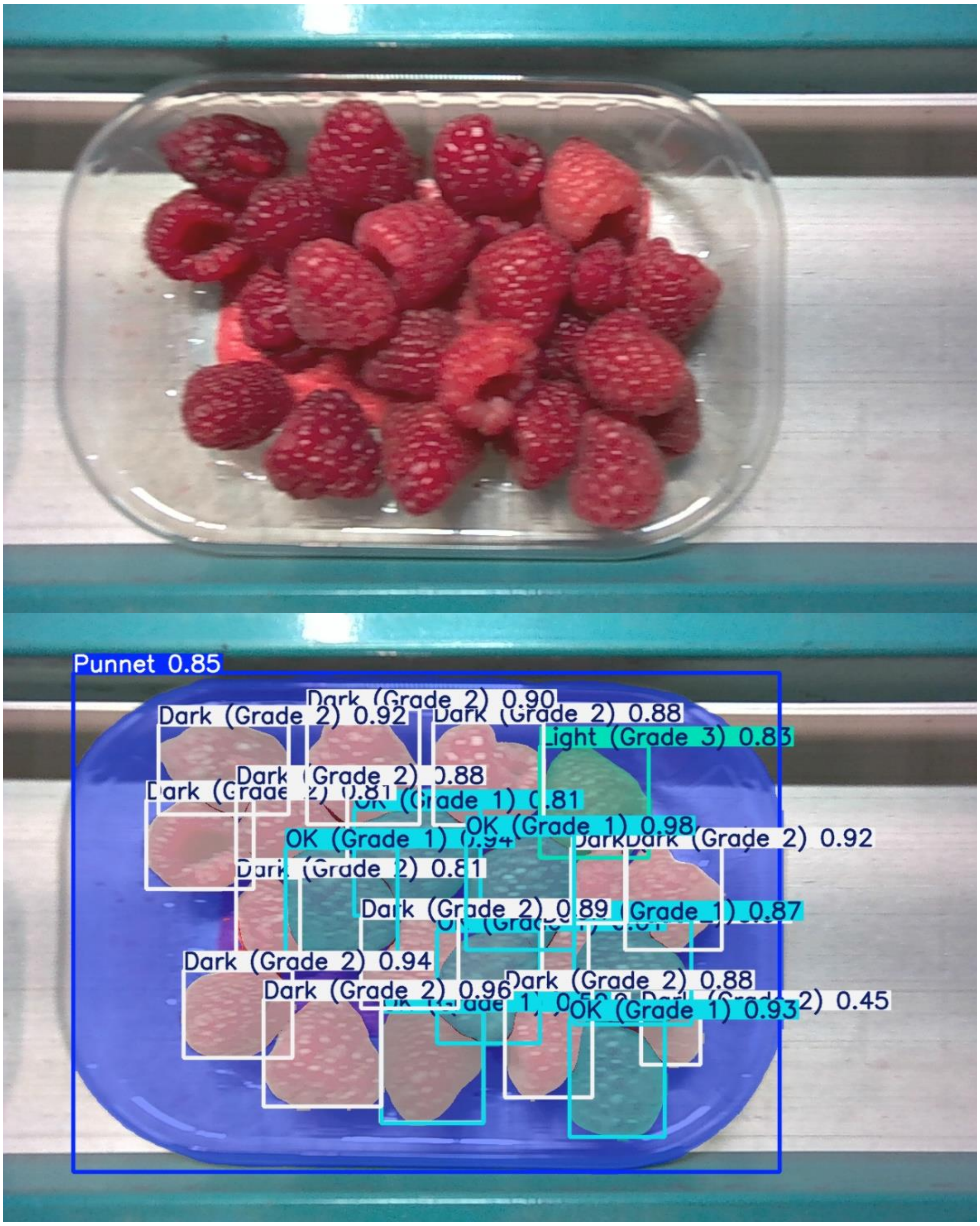


Figure 8: Punnet and raspberry before (top) and after (bottom) grading example.

2.2.1.3. Produmar

The steak grading task in this pilot consists of two steps, instance segmentation to isolate each steak in the scene and size-based grading. Steak instance segmentation was already concluded successfully in the previous release. Once the segmentation masks of the fish steaks are obtained from the RGB images (please refer to the first release for details), the depth images are leveraged to draw the grading criteria into classes A and B according to the size. To tackle the size-based grading into two classes (A and B), numerous criteria were explored such as maximum steak diameter, minimum steak diameter, diameter ratio, perimeter, area. However, the best results were observed when using the circumference and the area.

The circumference of a steak is inferred by summing up the distance between each two consecutive points lying along the contour of the predicted segmentation mask, while the surface is discerned directly by summing up the size of each pixel that belongs to the predicted segmentation mask. The distance between two points on the depth image and the size of the pixels therein can be determined by means of Intel® RealSense™ packages [5]. A large validation set that encompasses about 20800 steaks from grade A and 14500 from grade B was utilised. which were annotated by experts from **Produmar**. For each steak, the circumference and the area as indicated above were calculated, a histogram based on each criterion is built as depicted in **Figure 9**.

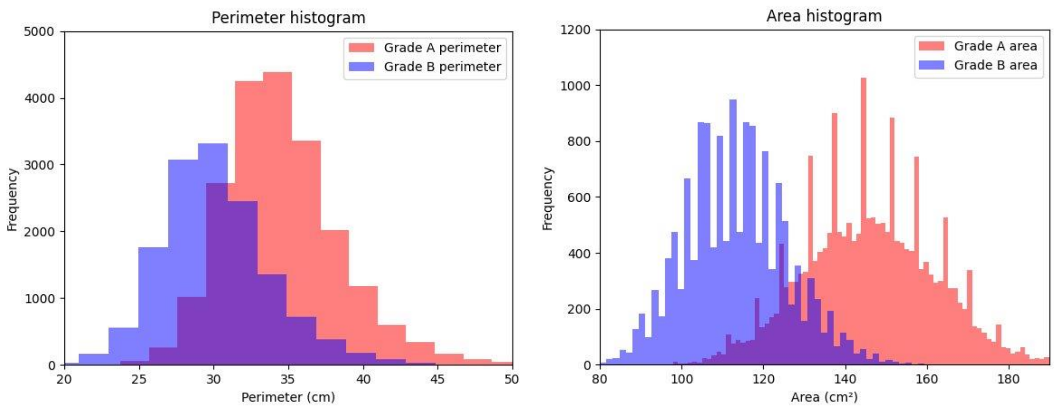


Figure 9: Histogram representation of steak circumference and area values.

It can be noted from **Figure 9** that the cross-grade intersection is larger when the circumference is adopted compared to the area criterion. This is directly reflected on the grading results given in **Table 1**, summarising the scores expressed in terms of the ratio of correctly classified steaks over the total number of steaks. In particular, the overall grading accuracy when considering the circumference amounts to 75.16%, which is boosted to 87.6% when the steak area is leveraged instead, suggesting a drastic improvement. It is to note that the grading thresholds regarding the circumference and the area were determined based on an empirical analysis and were set to 31 cm and 125 cm^2 respectively.

Table 1: Steak grading results (%).

Criterion	Grade A	Grade B	Overall
Circumference	84.16	62.29	75.16
Area	91.37	82.20	87.60

After the grading stage, the steaks are stored in a cold chamber. Eventually when there is an order, they are pulled out of the chamber and glazed on both sides for protection and sent to the packaging point. At this point, the vision system localises the steaks in 3D world and passes this information on to the robotic system (WP5) to proceed with pick and place operations (**Figure 10**).

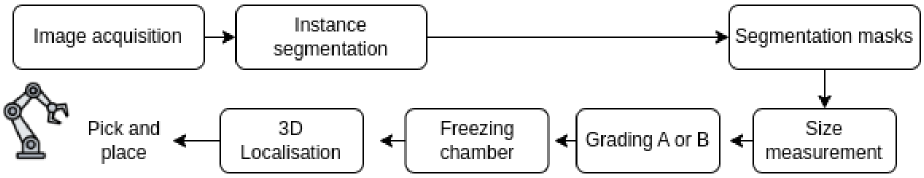
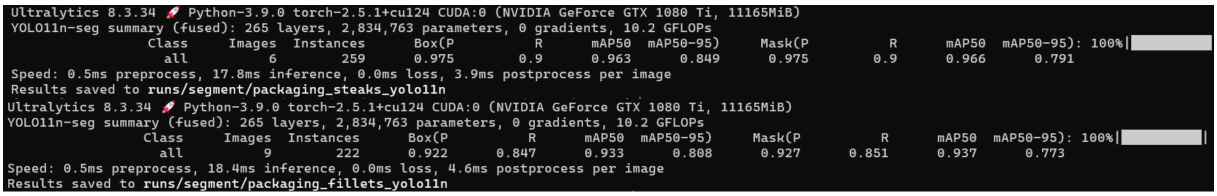


Figure 10: Pipeline of the grading and packaging system.

Besides the 3D localisation of the fish steaks, fish fillets 3D localisation is also considered. Therefore, instance segmentation of fish steaks and fillets based on YOLOv11 model is performed to determine the mask and isolate each steak/fillet sample in the scene. The segmentation model was trained on the training set, for 300 Epochs based on the pretrained yolo11n-seg backbone, and validated on the validation set (explained in the next section), and the results are reported in **Figure 11**.

Plausible scores have been achieved for steak and fillet segmentation, with mAP50 of 0.966 and 0.937, respectively. It is to mention that, since the vision system provides the 3D coordinates of fish samples to the robotic system, only one steak/fillet among all the detected steaks/fillets in the scene is localised and forwarded to the robotic system at a time. When the robotic arm completes the pick and place of a certain steak/fillet, another sample would have been already detected and localised for the next move. The 3D location of a steak/fillet can be discerned by means of Intel® RealSense™ packages [5] with respect to a marker that can be placed next to the conveyor. For instance, one can opt for ArUco markers [6]. Qualitative segmentation examples are given in **Figure 12**.



```
Ultralytics 8.3.34 🚀 Python-3.9.0 torch-2.5.1+cu124 CUDA:0 (NVIDIA GeForce GTX 1080 Ti, 11165MiB)
YOLO11n-seg summary (fused): 265 layers, 2,834,763 parameters, 0 gradients, 10.2 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|
                   all          6        259      0.975        0.9      0.963      0.849      0.975        0.9      0.966      0.791
Speed: 0.5ms preprocess, 17.8ms inference, 0.0ms loss, 3.9ms postprocess per image
Results saved to runs/segment/packaging_steaks_yolo11n
Ultralytics 8.3.34 🚀 Python-3.9.0 torch-2.5.1+cu124 CUDA:0 (NVIDIA GeForce GTX 1080 Ti, 11165MiB)
YOLO11n-seg summary (fused): 265 layers, 2,834,763 parameters, 0 gradients, 10.2 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100%|
                   all          9        222      0.922      0.847      0.933      0.808      0.927      0.851      0.937      0.773
Speed: 0.5ms preprocess, 18.4ms inference, 0.0ms loss, 4.6ms postprocess per image
Results saved to runs/segment/packaging_fillets_yolo11n
```

Figure 11: Instance segmentation scores of fish steaks (top) and fillets (bottom).

Figure 12: Instance segmentation examples of fish steaks (top) and fillets (bottom), before (left) and after (right) segmentation.

Although the grading and packaging tasks were concluded, it was decided with **Produmar** to address another non-grading task, where the head trimming lines of whole fish will be estimated by the vision system. This is to be tackled in the next step.

2.2.1.4. Marelec

For the **Marelec** use case, a grading problem of two classes is considered (refer to next section for major details). YOLOv11 was trained for 10 Epochs based on the pretrained yolo11n-seg backbone model on the training data and tested it on the validation set. The scores are given in **Figure 13**. The segmentation results are quite satisfactory given the evident contrast between the fillets and the background (the conveyor). For instance, a mAP50 of 0.994 for Grade A and 0.995 for Grade B are yielded. Qualitative segmentation and grading examples are depicted in **Figure 14**. After the grading step, 3D localisation is performed to enable further robotic handling (WP5). On this point, we follow the same procedure as explained in the **Produmar** use case above.

```
Ultralytics 8.3.33 🚀 Python-3.9.0 torch-2.0.1+cu117 CUDA:0 (NVIDIA GeForce GTX 1070 Ti, 8114MiB)
YOLO11n-seg summary (fused): 265 layers, 2,834,958 parameters, 0 gradients, 10.2 GFLOPs
val: Scanning /data/disk0/data/mekhalfi/AgileHand/codes/yolov8/datasets/Marelec/valid/labels.cache... 908 images, 0 backgrounds, 0 co
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95):
                   all        908        908      0.976      0.985      0.994      0.991      0.976      0.985      0.994      0.994
               grade A        430        430      0.952          1      0.994      0.991      0.952          1      0.994      0.994
               grade B        478        478          1      0.969      0.995      0.991          1      0.969      0.995      0.994
```

Figure 13: Instance segmentation scores of chicken fillets.

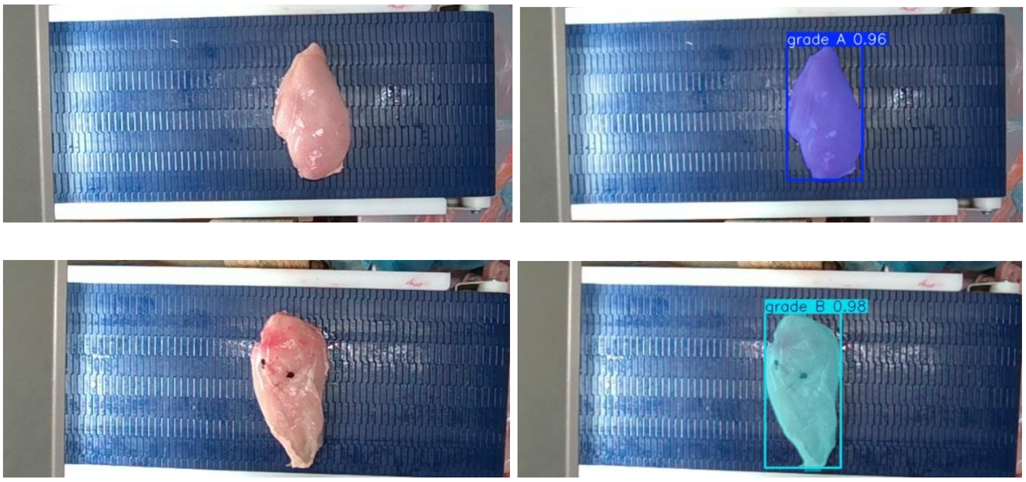


Figure 14: Instance segmentation examples of chicken breasts for grade A (top) and Grade B (bottom), before (left) and after (right) segmentation.

So far, the grading techniques for each pilot have been elaborated. Since our system is supposed to perform in real time, It is worth noting that our instance segmentation and grading solution exceeds 30 fps, which is beyond the requirements established by the pilots. Another essential property of any solution is scalability. In fact, our system can be easily scaled up or down and customised according to the grading needs.

2.2.2. Screenshot

For the user interface part and for each use case, a platform was developed in coordination with WP3. For instance, the grading results for each pilot are shared via a Structured Query Language (SQL) database in Python [7]. Afterwards, WP3 leverages the SQL data from WP4 and visualises it by means of Apache Superset™ in real time.

Below, screenshots regarding the login page are depicted, the pilots dashboard, and the Apache Superset™ data flow for each pilot, in **Figure 15**, **Figure 16** and **Figure 17**, respectively.

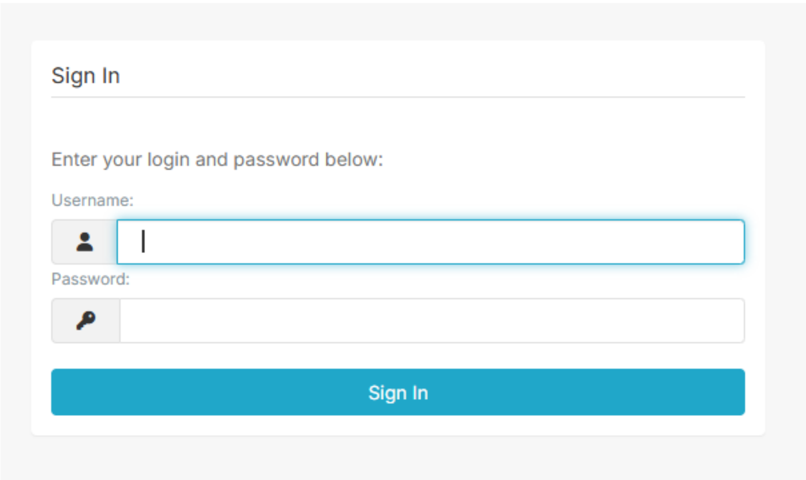


Figure 15: Login page to Apache Superset.

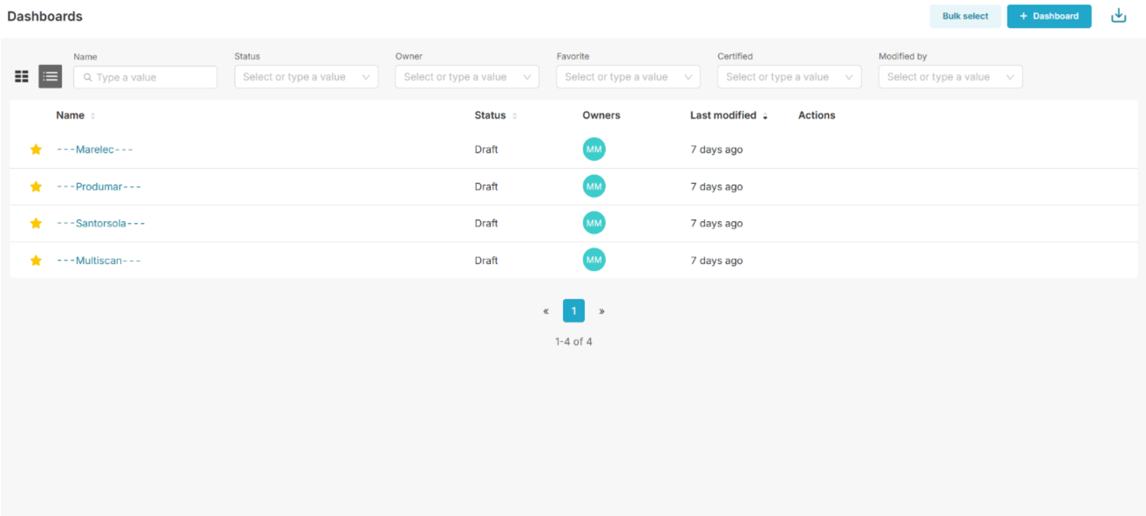
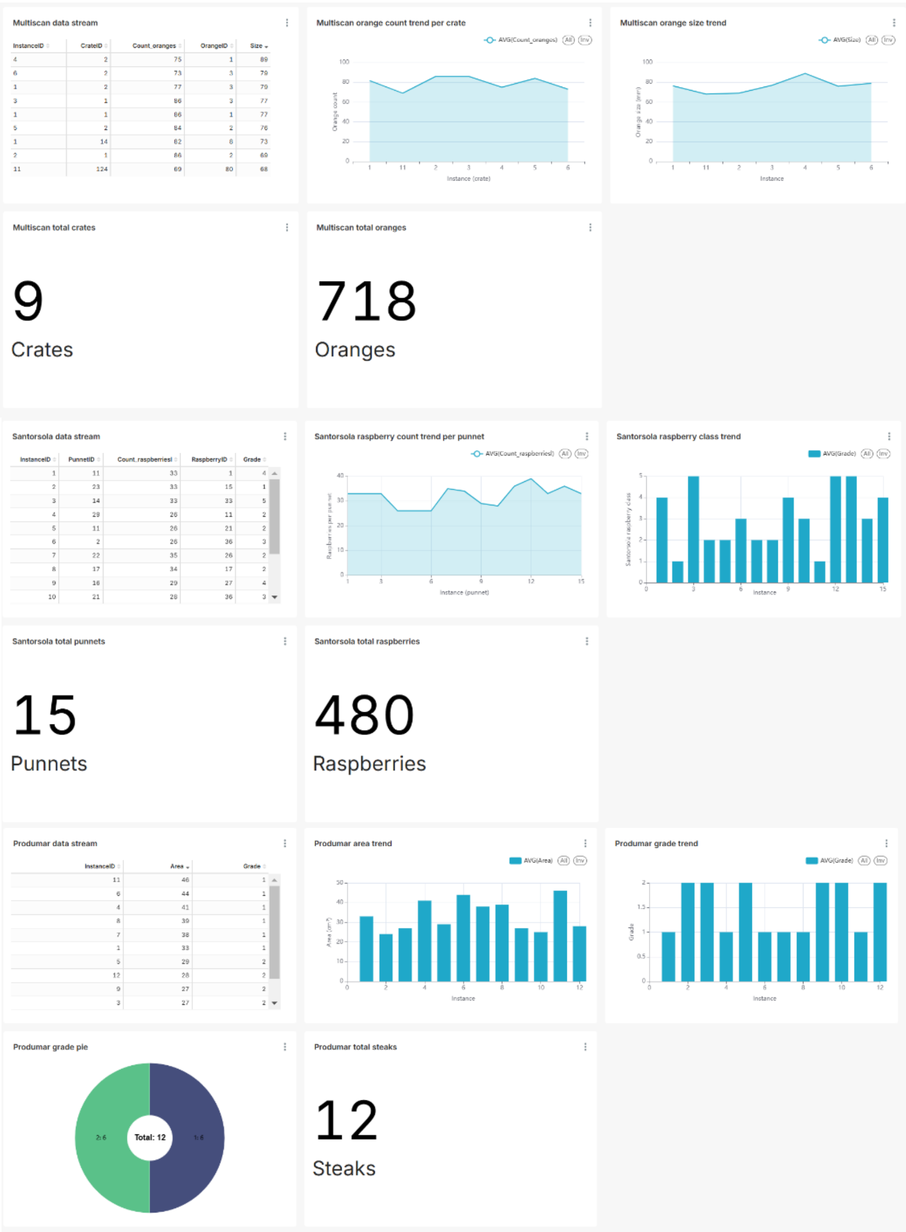


Figure 16: Pilots dashboard.



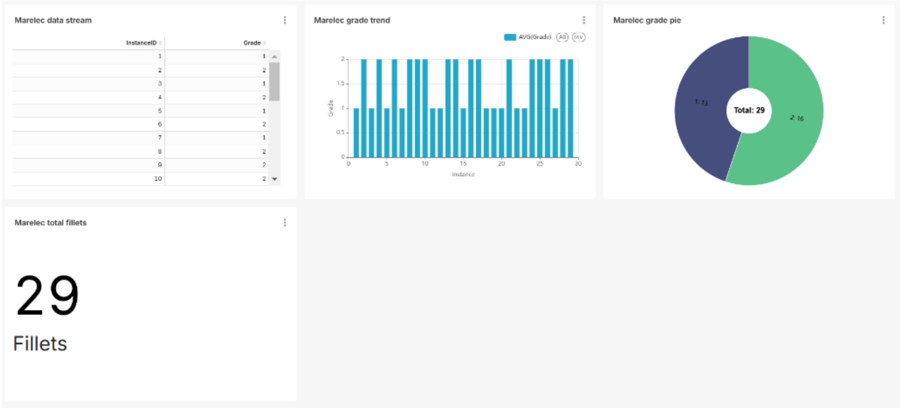


Figure 17: Data flow visualisation of Multiscan, Sant'Orsola, Produmar and Marelec.

3. AGILEHAND Data-Sets

3.1. Overview

Dataset acquisition (**Task 4.3**) and annotation is a pivotal step in **AGILEHAND** as we tailor data-driven techniques to address tasks such as fruit, fish, and chicken fillet segmentation, grading and localisation. In the previous release, the technical details regarding the pilots for which we acquired data were detailed, along with the upsides and the downsides where necessary. In this release, we build upon the conclusions that were drawn in the previous release.

3.2. Status

3.2.1. Release v2.0

3.2.1.1. Multiscan

In the previous release, the problem of fruit-level grading of oranges was tackled. In particular, a dataset that was collected by Multiscan Technologies, S.L (Pol. Ind. Els Algars C/ La Safor, 2 03820 Cocentaina, Alicante – Spain) via a Sony IMX429 camera was exploited.

Upon discussions with Multiscan, it was decided that a crate-level orange grading would benefit the grading process already existing in Multiscan. For instance, the crate-level analysis of oranges grades according to different criteria such as size, shape, and colour is prone to finetune the parameters of the grading machines according to the input batch (crate) and render the grading more reliable.

This implies the necessity to acquire a dataset that satisfies such requirements. The current plan is to acquire a dataset and finalise its annotation starting from month 24 (December 2024) when orange harvest season will already have begun. The annotation will be carried out using CVAT (Computer Vision Annotation Tool) online tool [8] and will take place right after the acquisition is achieved.

3.2.1.2. Sant'Orsola

Following the raspberries dataset acquisition, the images were annotated by means of CVAT tool [8]. First, the borders of the punnet were annotated. Second, the segment of each raspberry fruit inside the punnet was annotated. Third, experts from **Sant'Orsola** proceeded with the annotation of the class of each raspberry fruit. It is to recall that five classes are envisioned, namely Ok (Grade 1), Dark (Grade 2), Light (Grade 3), Second (Grade 4), Waste (Grade 5).

As per the statistics of the dataset, they are summarised in **Table 2**. In total, 200 raspberry punnets with varying number of fruits and class distribution were involved in the acquisition session. Afterwards, 160 punnets were considered for model training whilst the remainder (40 punnets) were left out for validation purposes, which account to roughly and 80%-20% training-validation split. As reported in **Table 2**, most of the fruits belong to Grade 2 followed by Grade 1, while Grades 4 and 5 feature the least samples.

Table 2: Dataset statistics per class of Sant'Orsola Pilot.

Class	Number of training samples	Number of validation samples
Punnet	160	40
Grade 1	773	205
Grade 2	2879	724
Grade 3	262	44
Grade 4	50	12
Grade 5	78	16
Total	4202	1041

3.2.1.3. Produmar

For this pilot, three tasks are envisioned. The first one regards the determination of the cutting line of the head part in a whole fish. The second one considers fish steak grading into two grades based on size. The last task addressed the problem of fish steak and fillet 3D localisation to enable the robotic solution (addressed in WP5) to handle them for the purpose of packaging.

Regarding the dataset acquisition part, it was already detailed in the first release along with the annotation process of the segmentation masks of the first (whole fish segmentation) and second task (steak segmentation). In this release, the remaining datasets of the first (head trimming lines) and third (steak and fillet packaging) tasks are further annotated. The statistics of the packaging datasets are provided in **Table 3**.

Table 3: Dataset statistics of Produmar packaging datasets.

Class	Number of training samples	Number of validation samples
Steaks	882	259
Fillets	1886	222

The head trimming task was already explained in the first release. At this stage, the annotation of the dataset for this task is concluded. Examples regarding the head trimming lines are given in **Figure 18**. The first row in this figure depicts a hake fish sample from Austral variety, the second row from Austral 2 variety, while the last row refers to New Zealand variety. These are samples of different hake fish that differ in terms of the sea zone where they were fished. In the images, we also show the coordinates of the two points that form the cutting line in the format (x1, y1, x2, y2), which are measured in pixels with respect to the top left corner of the image.

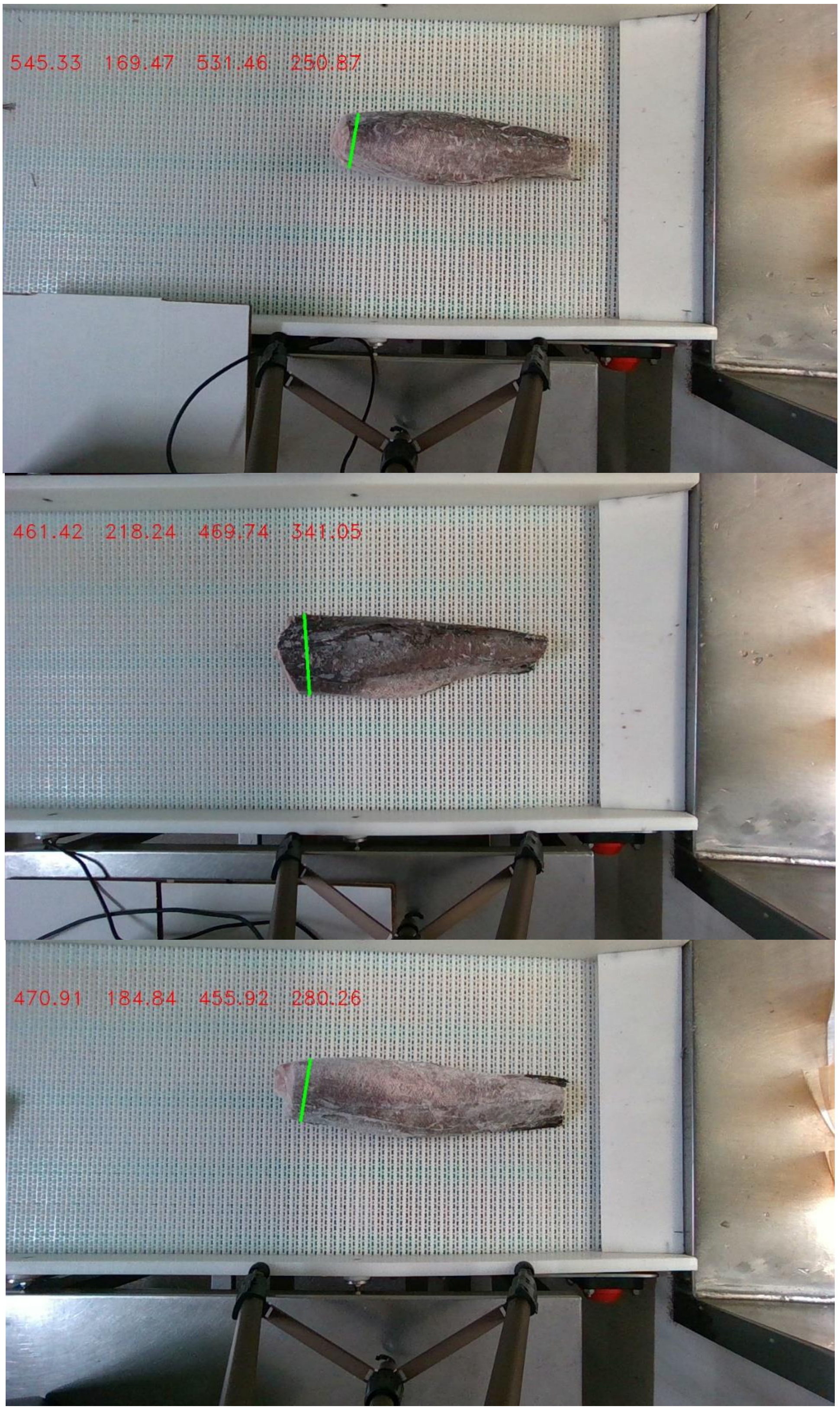


Figure 18: Examples of head trimming lines annotated by Produmar experts.

3.2.1.4. Marelec

For this pilot, a chicken fillet grading problem of two classes is envisioned, namely Grade A and Grade B. Grade A is superior in the sense that it shows a clean tissue with no defects, whilst Grade B often manifests blood clots and tears. To comprehend the visual differences, in **Figure 19** a tray from each grade is depicted.

Figure 19: Trays from both grades. Grade A (top) and Grade B (bottom).

In **Figure 20**, RGB and depth examples from both grades are displayed. Further, the statistics of the acquired datasets are summarised in **Table 4**.

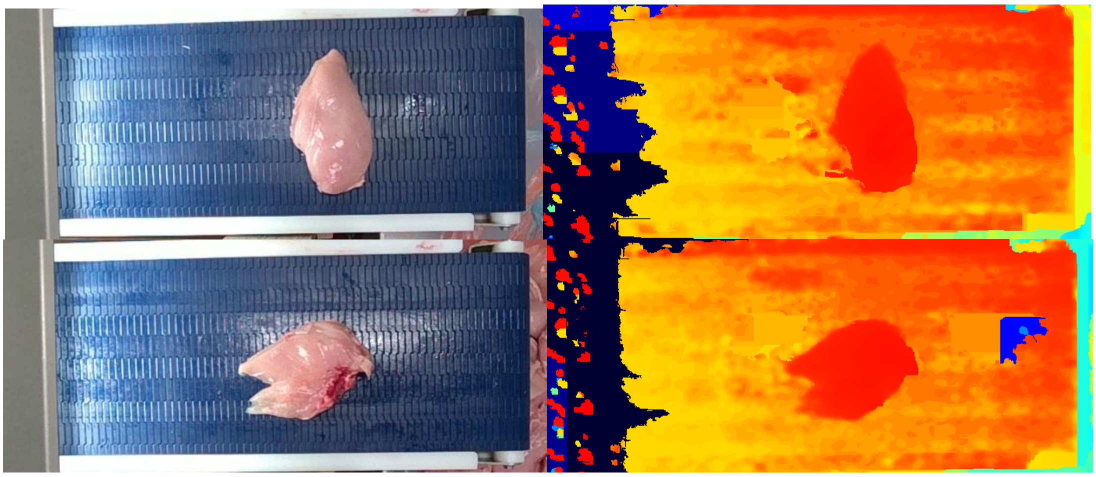


Figure 20: Examples from the acquired RGB (left) and depth (right) frames from Grade A (top) and Grade B (bottom).

Table 4: Dataset statistics per class of Marelec Pilot.

Class	Number of training samples	Number of validation samples
Grade A	596	430
Grade B	804	478
Total	1400	908

3.2.2. Screenshot

Regarding the use cases of Multiscan and Produmar, examples and screenshots have already been provided. In this section, screenshots from the acquisition process using CVAT tool [8] of **Sant'Orsola** and **Marelec** are displayed.

For **Sant'Orsola**, each class has a distinct annotation colour in CVAT [8] as shown in **Figure 21**. We also provide an example before (**Figure 22 top)** and after annotation (**Figure 22** bottom). The annotation colours viewed in **Figure 21** are the same colours featuring the contours of each fruit in **Figure 22**.

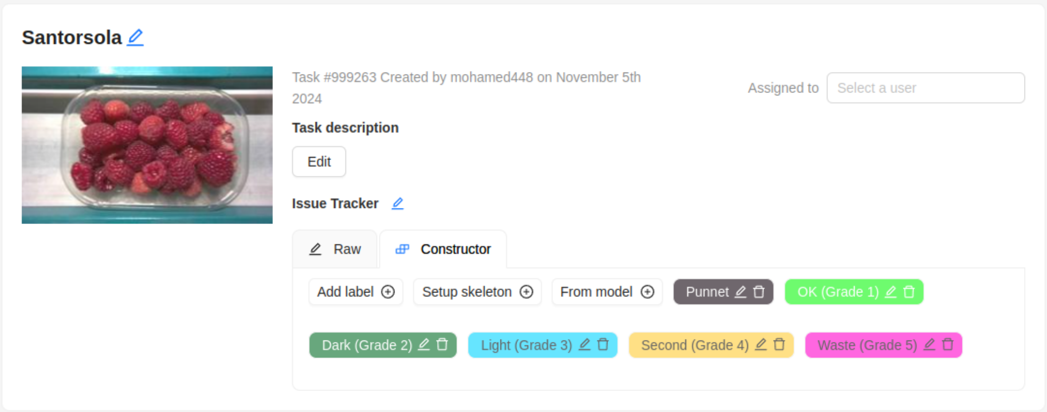


Figure 21: Grading classes for Sant'Orsola and their annotation colours on CVAT [8].

Figure 22: Illustration of the annotation process for Sant'Orsola using CVAT [8].

As per **Marelec**, two examples from the CVAT [8] tool are given, one from each class, in **Figure 23**. The classes here are colour-coded as well (Blue and Red for Grades A and B, respectively).

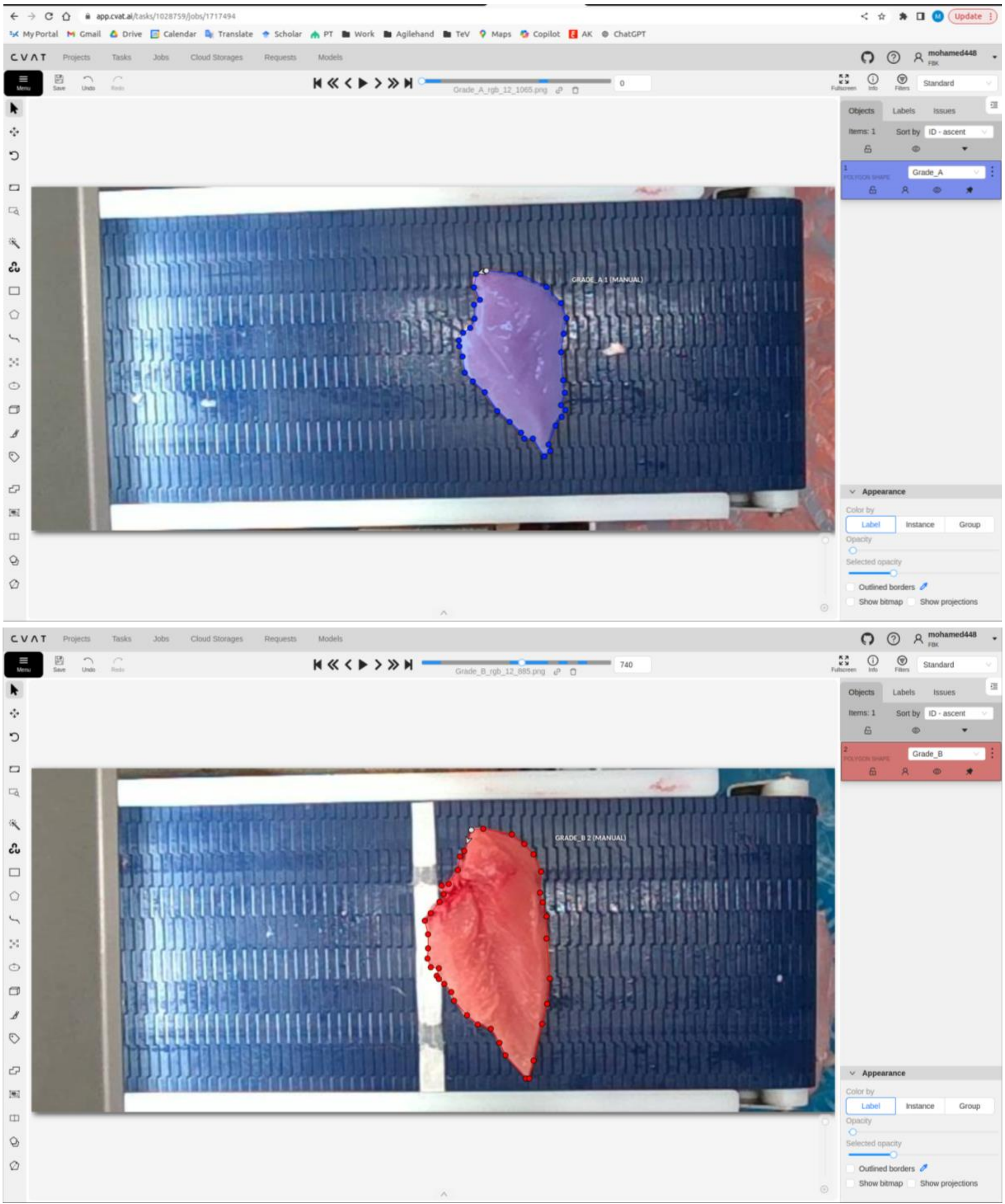


Figure 23: Annotation examples for Marelec pilot using CVAT [8]. Top: Grade A. Bottom: Grade B.

4. AGILEHAND Smart Sensing SUITE: Functional and Implementation Viewpoints V2

4.1. Functional Viewpoint

The details pertaining to the Functional Viewpoint of the solutions developed in **WP4** were detailed in Deliverable D4.1 and didn't require an update for this release.

4.2. Implementation Viewpoint

The details pertaining to the Implementation Viewpoint of the solutions developed in **WP4** were detailed in Deliverable D4.1 and didn't require an update for this release.

5. Conclusion

This document described the final development within WP4 of **AGILEHAND**. The upcoming period of the project will entail the wrap up and finalisation of the developed solutions, so they perform in real time at the premises of the pilots. This implies a continuous collaboration with other project partners to ensure smooth functioning and harmony between the individual WP solutions.

6. References

[1] Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. Computer Science Review, 40, 100379.

[2] https://github.com/ultralytics/ultralytics

[3] Padilla, R., Netto, S. L., & Da Silva, E. A. (2020, July). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE.

[4] Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & Da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics, 10(3), 279.

[5] https://dev.intelrealsense.com/docs/

[6] Kedilioglu, O., Bocco, T. M., Landesberger, M., Rizzo, A., & Franke, J. (2021, October). ArUcoE: enhanced ArUco marker. In 2021 21st International Conference on Control, Automation and Systems (ICCAS) (pp. 878-881). IEEE.

[7] https://www.w3schools.com/python/python_mysql_getstarted.asp

[8] Accessible at: https://app.cvat.ai/