Ref. Ares(2023)4546155 - 30/06/2023

This project is funded by the European Union
G.A No 101016906

Project Number: 101016906
Start Date of Project: 2021/01/01
Duration: 48 months

Deliverable D4.2 V1.0

Agronomic Quality Assessment

Dissemination level	PUBLIC
Submission Date	2023-06-30
Work Package	WP4
Task	T4:3
Type	Report
Version	1.0
Author	Thomas Ciarfuglia, Leonardo Saraceni, Ionut Motoi, Mulham Fawakherji
Approved by	CONSORTIUM

DISCLAIMER:
The sole responsibility for the content of this deliverable rests with the authors. It does not necessarily reflect the opinion of the European Union. Neither the Directorate-General for Communications Networks, Content and Technology, Artificial Intelligence and Digital Industry nor the European Commission are responsible for any use that may be made of the information

This project is funded by the European Union G.A No 101016906

Executive Summary

This document summarises the design and research questions and the relative findings arising from task 4.3 “Agronomic Quality Assessment”. This task has the goal of designing perception systems that can be integrated into the CANOPIES experimental robotic platform to enable the agronomic operations related to harvesting, and therefore giving the foundation for the Human-Robot Interaction (HRI) tasks that exist on a higher conceptual level.

The main quality parameters have been identified in WP2 and summarised in deliverable D2.3, together with the relevant KPIs. Among those parameters, some were chosen and have been the object of data collection and algorithm design and validation. These are:

- Colour level
- Brix° (sugar content)
- Damages
- Bunch and berry size

Other parameters, such as bunch density, while interesting, were considered in later discussions with the agronomic partners of AGRIMESSINA as secondary and not relevant to the harvesting decisions.

The design of an estimator for each one of the four quality parameters chosen is a challenging effort per se, at the same time an overarching Quality Estimation Module (QEM) had to be developed to have better integration with the overall system and to better support the decision-making processes that are considered in other tasks. In this report, we describe each estimator, data collection, algorithmic solutions and experimental validation, and also describe the overall QEM system architecture and functionalities.

Table of Content

1 Introduction 6
2 System Overview 8
2.1 System Architecture 8
2.1.1 Tools 9
2.2 System requirements 9
2.3 Remarks on data acquisition and elaboration 11
2.4 Sections outline 11
3 Data Collection 13
3.1 Introduction 13
3.2 Experimental Field 13
3.3 Devices and Software Tools 13
3.3.1 Devices 13
3.3.2 Labelling Tools 14
3.4 Dataset Description 14
3.4.1 Segmentation and Detection Data 19
3.4.2 Anomaly Detection Data 19
3.4.3 Color and Brix Data 22
4 Quality Estimators 31
4.1 Anomaly Detection Module 31
4.2 Color and Brix Estimation Module 32
4.2.1 Hand-crafted features for brix estimation 32
4.2.2 Early colour classification with MobileNetV3 34
4.2.3 Representation learning with multi-task architecture 34
4.3 Size and Shape Estimation Module 35
4.4 Detection and Tracking 37
5 Experiments and Discussion 40
5.1 Anomaly detection validation 40

5.2 Brix and colour estimation validation 41
5.2.1 Exploratory data analysis: 42
5.2.2 Color label consistency 43
5.2.3 Baseline colour estimator validation 44
5.2.4 Histogram feature validation 47
5.2.5 Multi-task 50
5.3 Size and shape estimation validation 53
6 Conclusion 56
7 Bibliography 57

Abbreviations and Acronyms

WP	Work Package
QEM	Quality Estimation Module
KPI	Key Performance Indicator
GUI	Graphical User Interface
HRI	Human-Robot interaction
RGB	Red Green Blue (referring to the camera channels)
RGB-D	Red Green Blue + Depth (referring to the camera channels)
ROS	Robot Operating System

This project is funded by the European Union G.A No 101016906

1 Introduction

Agronomic-oriented perception in the context of the CANOPIES project is the set of perception capabilities that the robot manipulator should have in order to interact and be of help to a human farmer. The project envisions a complex set of interactions, between humans and robots, that revolves around the harvesting task. While basic perception involves the capacity to "see" the environment, the obstacle, the landmarks and so on, and thus is a common feature of many work packages, perceiving something for quality poses a very different and challenging task, as it will be clarified in the following paragraphs.

Quality estimation through perception is the objective of Task T4.3, and the boundaries of its characteristics have been discussed by consortium partners during WP2. According to the results of that work package, distilled in deliverables D2.2 and especially D2.3, there are a number of quality parameters that are of special importance in table grape cultivation:

- Skin colour
- Sugar level and acidity
- Bunch density
- Berry size
- Defects

Each one of these parameters is in itself a different problem to tackle. While there are synergies between some of the strategies for data collection, and for model definition, that can be leveraged, each of the aforementioned quality parameters has its own design process. During the early stages of WP4, while possible sensors and solutions were explored, and state-of-the-art was assessed for fruitful directions, the general approaches to the estimation of the quality parameters have been delineated:

Skin colour: This parameter is one of the most important since it is one that even an untrained person can judge. Estimating colour means classifying some colour categories since having a continuous "colour" index did not seem very practical. However, finding a good definition of colour classes is not trivial. Following AGRIMESSINA's suggestion, we used a colour chart provided by them to categorize the colour quality levels. One of the challenges that emerged very soon is that the colour categories have a good deal of interpretation and arbitrariness, that adds up to label noise. The question of whether this noise is too high to reach reasonable solutions was an early consideration of this sub-task.

Sugar level and acidity: the consortium partners agreed very early in the project definition on focusing on the sugar level content, a.k.a. Brix degree (brix°). Even with this choice, this parameter is one of the most difficult to estimate, since it is an internal physical characteristic of table grape berries, and it has no direct correlation with external appearance. In the early stages of the task, the use of multi-spectral cameras has been considered. However, after a market study and the analysis of the state-of-the-art, it was clear that there was no easy solution for a robot that has to work in daylight, with unknown illumination conditions. Academic literature using multispectral and hyper-spectral cameras would do it with controlled artificial light, or in laboratory conditions. Given that, and the fact that human workers are able to assess ripeness through external examination, even with a good degree of uncertainty, we decided to test if there was enough correlation between colour and brix°, to be able to estimate it with only RGB cameras. As this document will describe, this hypothesis proved to be true, and an estimator based on appearance was designed.

Bunch density: this was a parameter of interest only in the first stages of the task. After investigating with AGRIMESSINA the possible variation of the parameter, it was apparent that bunch density rarely impacts harvesting decisions, i.e., only in very extreme cases the bunch is not collected for density problems. These problems could be considered anomalous external conditions, so we decided not to have a specific estimator for this parameter, but to use defect estimation and size estimation as the possible means to assess critical bunch density situations.
Berry size: it was clear from the early stages of the task that, given the presence of a stereo camera on the robotic manipulator, the berry size estimation pipeline could use geometric algorithms instead of data-driven ones.
Defects: this parameter is a catch-all word for any possible anomalous situation that could lower the value of the table grape as a product, or even worse, detrimental to the well-being of the vine tree and its neighbours. Since we are considering very different situations, as described in deliverable D2.3, we followed the idea of framing the problem as an anomaly detection problem.
To add further complexity to the picture, quality decisions should put together the information from all these quality estimators, and be able to provide the farmer with a simple assessment that can be operated in a binary harvesting decision. For this reason, to reach the goal of the task, an overall Quality Estimation Module (QEM) has been designed, encapsulating all the four aforementioned estimators, and working as a ROS service providing functionalities to higher-level algorithms for human-robot interaction.
While the development of the estimators and of the QEM has been a highly non-linear process, in this document we try to explain data collection, developed methodologies and experimental validation in a systematic way. Section 2 will therefore introduce the overall QEM system, Section 3 will describe the data collection for each of the four estimators, plus the preliminary data collection and labelling for detection and segmentation. Section 4 will introduce the methodologies developed and Section 5 will explain the experimental validation of both data and models. Finally, Section 6 will draw conclusions and describe future work.

2 System Overview

2.1 System Architecture

To have a comprehensive understanding of task T4.3 it can be useful to start with a high-level description of the QEM. While the system was in early development, it wasn't clear if quality estimation would have been inter-operating with other ROS modules developed in other WP. The team of researchers working on it worked on the architecture and code implementation using an agile project management (SCRUM with modifications) style and opted for a fast-prototyping approach. The first prototype helped in defining the correct interfaces and relative responsibilities of nodes and classes.

In the current version, the QEM is designed to work as a service. This is mainly due to computational constraints that we foresee in the early stages of development. Since estimation algorithms are often based on deep learning methods, and knowing that GPUs and GPU memory are a very limited resource on the manipulator, the “on demand” operation was the natural architectural choice. This choice has some effect on the design of the demos and of the human-robot interactions, as detection and safety modules will run continuously, while agronomic quality perception will run only when a direct request will be made.

The general QEM ROS system architecture is depicted in Fig. 1. The diagram shows the ROS nodes responsible for perception and that are used also for harvesting operations that are required for the QEM functioning. The basic functioning requires a table grape detector to crop images collected from the camera and organise them in batches of the same size. We opted for working with fixed-size input images, without padding. Tracking of grape instances is not required for QEM functioning, but we hypothesised that it would be a possible need to integrate multiple quality estimations of the same grape bunches, with images taken from different points of view. For this reason, detection and tracking were developed as part of the QEM codebase.

Any kind of harvesting decision node is going to be external to the QEM. This means that the QEM has to provide all the information needed for both simple and complex decision-making systems. Where it is appropriate, the prediction is given together with some confidence score on the estimate.

Another architectural requirement for the QEM is that the sensor should provide the point cloud aligned with the RGB image since berry and bunch size estimation and distance computation are based on it.

The QEM is not designed to work in real-time, even if some sub-modules are able to. The main limitation to real-time performance is again GPU memory availability on the manipulator, whose amount is not clear at the time of the writing of this report. Switching between different models required for different tasks, not only the QEM sub-modules, could limit the possibility of working in real-time. This issue will be explored in the WP9 on field validation.

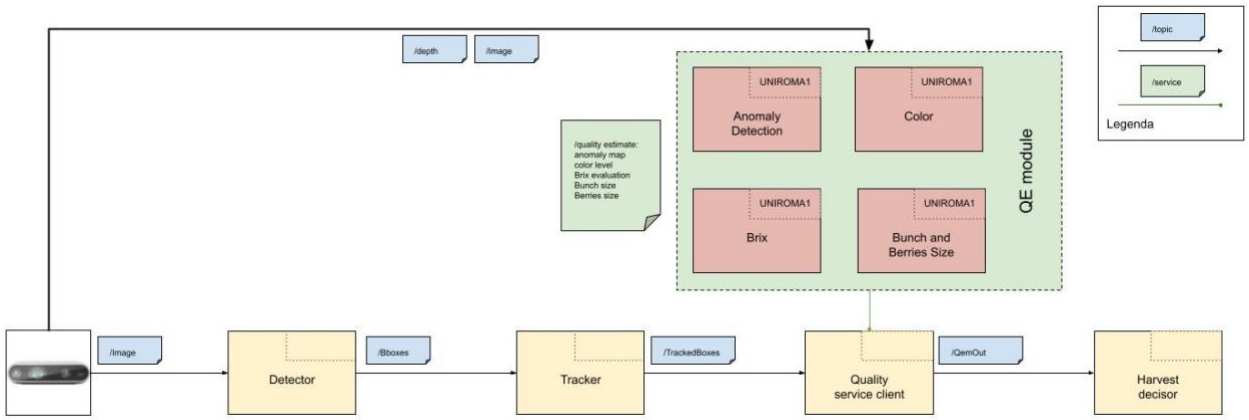

Figure 1 This diagram shows the current architecture of the QEM and some of the ROS nodes that have been developed for its working. All the blocks in the diagram are ROS nodes. The QEM code base contains the four main quality estimators, a YOLO based detection module, and a trackier.

2.1.1 Tools

Defining the four estimators sub-modules is a challenging scientific endeavour, but developing the QEM is also a considerable engineering effort. For this reason, it was very important for the team working on it to use the right tools and procedures. This subsection briefly describes these aspects:

Code versioning: as per common practice, code versioning was organised using git and GitHub for collaboration.

Data versioning: an increasingly important aspect in MLOps is data version control since different iterations of datasets can affect the outcomes of the experiment, as much, if not more, than the modelling choices themselves. We considered a number of different version control systems for data, and the shortlist was composed of Weight & Biases[9] tools and DVC[10]. In the end, W&B was chosen for data versioning since we also used it for model visualisation.

Project management tools: as mentioned before, an agile project execution was followed. To support this we initially used Jira[11], but found it lacking many basic features. In the end, we moved to YouTrack[12] as the main issue-tracking system.

2.2 System requirements

In this Section, we summarise the requirements extracted from deliverable D2.3. The underlying hypothesis for the definition of the specification given in that document is that the robot should be able to have minimum evaluation performances close to human judgement. This was roughly estimated by AGRIMESSINA experts to be around 80% of final harvesting accuracy, meaning that 20% of the grape bunches collected by human workers do not pass the quality control criteria for

This project is funded by the European Union G.A No 101016906

distribution. It is not easy to translate this final performance figure to specific performance metrics for each one of the estimation sub-modules. In the following the metrics defined in deliverable D2.3 are described and commented:

Berry size estimation: *achieve a mean absolute error (MAE) less than 30% of berry size on benchmark data*. Pizzutello Nero does not have spherical berries, so the measurements that were considered are the size of the bounding box as an approximation of the length of the berry, and the width as an approximation of the diameter. The critical aspect with respect to the harvesting is not to distinguish between small variations in berry size but to detect under-developed berries, much smaller than the standard ones.

Colour classification: *achieve at least 80% of accuracy on grape bunch colour classification on benchmark data*. This metric was defined before the colour classification scale was introduced during the development of this task, described in Section 3.4.3. It refers to the final harvesting accuracy, while the estimator developed in the end is able to classify different degrees of colour quality. So, balanced accuracy on the multi-class problem should not be confused with the final harvesting accuracy based on colour alone.

Sugar content estimation: *achieve an MAE less than 5 °Brix with respect to the measured value of °Brix sugar level using visual sensors on benchmark data*. This error threshold was very high, both because human workers have actually hard time guessing sugar content with their own eyes, and because estimating brix* in the wild with normal RGB cameras is an open research question, more than an engineering goal. The high MAE target was taking into account these uncertainties.

Density estimation: *achieve at least 80% of accuracy on bunch density estimation on benchmark data*. As we mentioned in the previous section, discussing the possible taxonomy of grape bunch density with AGRIMESSINA, it became clear that very rarely bunch density becomes an issue for harvesting decisions. It impacts the quality of the bunch, but on the later stage of production and distribution, not on the field. For this reason, this performance metric became irrelevant for the QEM development, since bunches that should be discarded due to bunch density are also defective for size estimation, or other factors.

Defects estimation: *achieve at least 80% of accuracy on defects and sickness identification on benchmark data*. Again, the accuracy is on the harvesting decision, while, depending on the method used for defect estimation, other more suitable metrics have been used for evaluation.

2.3 Remarks on data acquisition and elaboration

To be able to follow the description given in the rest of the document, it is necessary to stress that the data acquisition campaigns for the four estimators were very interleaved. The document describes the data acquisition and processing for each estimation problem as a separate acquisition, but the reader should keep in mind that the field trips were the same, and that, for example, the same video sequence o set of static images could have been used to provide data for detection, brix estimation, colour estimation or to provide examples of anomalous grape bunches. This idea is simplified in Fig. 2.

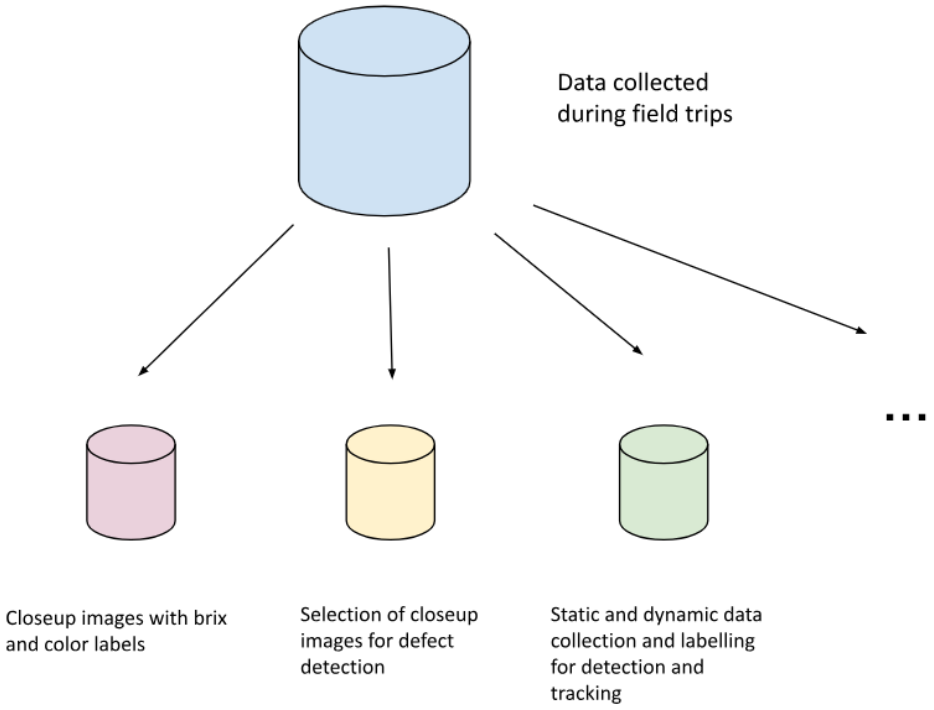

Figure 2 Data collection and processing for the different sub-modules was an interleaved process, with both overlaps and specificities between the processes.

Finally, it is also important to remind the reader that the seasonality of data collection has been one of the biggest challenges and constraints to the task development, as after a data acquisition season (September-October), any correction and modification to the procedure and data collected has to wait twelve months to be implemented. For this reason, if some of the choices made in the design of the estimators could appear unusual, take into account the scarcity of data and the limitations on amending data collection.

2.4 Sections outline

The rest of the document is organised as follows:

Section 3 describes the data acquisition and processing.

Section 4 describes the modelling of the estimators.

This project is funded by the European Union G.A No 101016906

Section 5 describes the experimental validation of both data (if needed) and models.

Section 6 draws the conclusions.

Each Section is roughly organised with the same pattern, describing the core topic of the section from the point of view of each estimation problem. The consideration given in this Section should be always kept in mind to have a correct view of the whole task development process.

3 Data Collection

3.1 Introduction

Each of the four quality parameters has slightly different requirements for data collection, however, they are also connected by common features, such as the field characteristics and the sensor used to collect the images. In the following we initially describe the common features of the data object (the grape varieties), context (the experimental field in Aprila) and sensors. The specific requirements of data collection for each estimation problem will be addressed in the corresponding section.

3.2 Experimental Field

Figure 3 Satellite view of the experimental field

The experimental field is located in southern Lazio (Italy). The vineyard is composed of two plots approximately 114 m x 51 m (0,58 ha) and 122 m x 48 m (0,58 ha). Vineyards are structured as a traditional trellis system called Tendone with a wide distance between each plant, 3 x 3 m2. Plantations are all older than 3 years and so in full production and health, thus representing a typical working condition for the validation of agronomic activities such as fruit harvesting or vine pruning. All structures are traditionally covered with plastic and a net to protect grapes from rain and hail. The average extension of each plot is around 1 hectare and dimensions (length and width) are on average between 25 m and 50 m according to plot extension and geometry. The selected vineyard in Aprilia has currently four different table grape varieties which are described in the following: White Pizzutello, Black Pizzutello, Red Globe and Black Magic. Fig. 3 shows images of the experimental field as well as the approximate extension of each grape variety in the vineyard. Of the four varieties that were present in the vineyard, Black Magic was of very low quality and thus untended by the field owner. White Pizzutello is identical in shape to the Black one, and the latter has the same colour as the former when not ripe. Together, white and black Pizzutello are a peculiar variety of the Lazio Region and present the highest variability in shape and colour with respect to standard rounded berry variants. For these reasons, while we collected images of all the varieties, the consortium decided to concentrate all the demonstrations on the Black Pizzutello variety, therefore we finally concentrated most of our data labelling effort only on Black Pizzutello.

3.3 Devices and Software Tools

3.3.1 Devices

During the first two summer seasons the project covered so far a number of different sensors have been used to collect different kinds of data. Standard camera sensors together with the Realsense D435i were employed in both seasons, in order to be able to have an estimate of the covariate shift

due to different devices by the learning algorithms, particularly for the colour and brix estimation problems.

2021:

- Smartphone camera (Moto G8)
- Reflex Camera
- Realsense d435i
- Realsense d415

2022:

- Smartphone camera (Realmi)
- Realsense d435i

The d435i images can be considered our final target images, since this device is the camera type that is installed on the manipulator, and all algorithms will be finally evaluated only on this device during the field validation runs of WP9.

3.3.2 Labelling Tools

For this task, two main software packages were used to label the images:

Innotescus[2]: This labelling tool has been used in the 2021 campaign, particularly for the initial labelling of the detector and segmentation algorithms that were also part of tasks T4.1 and T4.2. Unfortunately, Innotescus was a start-up and they shut down the business shortly after the first paper related to weakly and self-supervised learning was published. The data labelled with that software has been backed up and is still usable, nonetheless.

CVAT[3]: this is an open-source annotation tool that allows for complex annotations such as instance-level bounding boxes and segmentation masks with attributes. Of particular interest for this task, was the feature of interpolating bounding boxes across multiple frames of a video sequence, which has been widely used for the creation of the validation data for the tracking algorithms described in Section 4.4.

3.4 Dataset Description

As mentioned before, there are different kinds of data collected for different purposes and with different devices. The organisation and cataloguing of this huge amount of raw data, and the preprocessing and labelling required a considerable amount of work. In this section, a general description of the data collected is given. More specific considerations will be given in the following sections.

Static images:

Static images have been collected for the detection, segmentation, colour, brix and anomaly detection sub-tasks. These images have been collected using both smartphone cameras, a reflex

camera and the d435i camera. In the very first data acquisition campaigns, also a d415 camera was used, but we dropped its use since the final device on the robot was chosen to be the d435i.

Most of the static images were collected together with the brix measurements, which will be described in Section 3.4.3. However, some images of rotten and sick grape bunches were also collected for the specific purpose of having negative examples for the anomaly detection sub-task.

ROS bag files:

ROS bagfiles[4] are a file format for sequences of messages passed through the ROS network between applications. In our case, they are a convenient way to record sequences of images collected by the d435i device, together with all the parameters that are required for the functioning of the device. Since we collected the bagfiles at different moments, the different characteristics of their contents reflect our understanding of what was needed at a certain time. Table 1 with a description of the video sequences is given. This taxonomy was used to identify the best sequences to be used as a validation set for our algorithms.

All these sequences have been extracted from the bagfiles using a specific tool, and everything was saved on Google Drive until a final open-source publication of the data will be done. We stress that the huge amount of data collected is an asset of the project, and that in the following system description only a fraction of it has been used. Future work will be able to leverage more of these data.

Table 1: Taxonomy of the bagfiles recorded from 2021 and 2022 field visits. The main parameters of interest are the general camera motion (left, right, closeup, forward), the position of the sun with respect to the camera, the type of data recorded (only RGB, or also IR, Depth and IMU), the use of colour correction card and the general characteristics of colour and anomaly severity, if any. Some cells have different colours to show some specific properties. In particular, the green ones have been selected to be the test set for the tracking algorithms that are described in Section 4.4

Name	Date	Anomaly	Anomaly type	Min color level	Max color level	Sun position	Moving	IR	IMU	Card
20210930_155706.bag	30 Sep 2021	no	-	6	4	back	left	yes	yes	yes
20210930_155841.bag	30 Sep 2021	yes?	small	6	3	front	right	yes	yes	yes
20210930_160215.bag	30 Sep 2021	no	-	6	6	back	closeup	yes	yes	no
20210930_160007.bag	30 Sep 2021	?		6	1	back	right	yes	yes	yes
20220808_122420.bag	8 Aug 2022	no	-	6	4	front	right	no	no	no
20220808_122241.bag	8 Aug 2022	no	-	6	5	back	left	no	no	no
20220808_122602.bag	8 Aug 2022	no	-	verde	medio	front	forward, closeups	no	no	
20220808_123000.bag	8 Aug 2022	no	-	green	alto	front	left	no	no	
20220822_093202.bag	22 Aug 2022	no		6	5		no	no	no	no
20220822_093434.bag	22 Aug 2022	no	-	6	5		left	no	no	no
20220822_093628.bag	22 Aug 2022	no	-	6	4		right	no	no	no
20220822_093927.bag	22 Aug 2022	no	-	6	4		right	no	no	no
20220822_094153.bag	22 Aug 2022	no	-	6	5		U shape	no	no	no
20220822_094349.bag	22 Aug 2022	no	-	6	4		U shape	no	no	no
20220822_101300.bag	22 Aug 2022	no	-	6	5		no	no	no	yes
20220822_094850.bag	22 Aug 2022	no					left	no	no	no
20220901_143110.bag	1 Sep 2022	no	-	6	1	back	left	yes	yes	no

20220901_142936.bag	1 Sep 2022	?		6	1	strong back and front	circle/closeup	yes	yes	no
20220901_142743.bag	1 Sep 2022	yes	small	6	2	strong back and front	U shape	yes	yes	no
20220907_154427.bag	7 Sep 2022	yes	bunch dissecation	6	1	back	left	no	no	no
20220907_153958.bag	7 Sep 2022	yes		5	1	back	left	no	no	no
20220907_154253.bag	7 Sep 2022	yes		5	1	back	U shape	no	no	no
20220907_094532.bag	7 Sep 2022	yes					closeup	no	no	no
20220907_093104.bag	7 Sep 2022	no					U sh	no	no	no
20220907_093312.bag	7 Sep 2022	yes						no	no	no
20220907_093734.bag	7 Sep 2022	yes					U sh	no	no	no
20220907_094135.bag	7 Sep 2022	yes					u sh	no	no	no
20220907_094210.bag	7 Sep 2022	yes					closeup	no	no	no
20220907_094248.bag	7 Sep 2022	yes					closeup	no	no	no
20220907_094621.bag	7 Sep 2022	yes					closeup	no	no	no
20220808_123157.bag	8 Aug 2022	Red Globe								
20220808_123331.bag	8 Aug 2022	Red Globe								
20220808_124833.bag	8 Aug 2022	Black magic								
20220808_125053.bag	8 Aug 2022	Black magic								
20210930_160434.bag	30 Sep 2021	no	-	6	6	all around	closeup	yes	yes	no
20210930_160626.bag	30 Sep 2021	no	-	6	6	all around	closeup	yes	yes	no
20210930_160910.bag	30 Sep 2021	no	-	7	7	all around	closeup	yes	yes	no

20210930_161059.bag	30 Sep 2021	no	-	7	7	all around	closeup	yes	yes	no
20210930_161318.bag	30 Sep 2021	no	-	7	7	all around	closeup	yes	yes	no
20210930_161415.bag	30 Sep 2021	no	-	7	7	all around	closeup	yes	yes	no
20210930_161518.bag	30 Sep 2021	no	-	7	7	all around	closeup	yes	yes	no

3.4.1 Segmentation and Detection Data

To be able to work on quality estimation in parallel to task T4.1 about detection of grape bunches, and initial work to have basic detection and segmentation functionalities has been done. From the point of view of data collection, we labelled some data for detection and segmentation. The data collected for this purpose has been also used for the semi-supervised pseudo-label generation described in [5] and summarised in deliverable D4.1. This fraction of data has the following characteristics.

Since detection and tracking are closely related issues, we selected some HD (1280×720) videos at 10 Hz with a total of 1469 frames. A short segment of 10 seconds has been labelled for test use in the case of the tracking algorithm evaluation, while the rest has been used without labelling thanks to the semi-supervised nature of the system. Note that, since the data was also collected for other purposes, the images contain a neutral grey card for white balance purposes. We also acquired and selected a second kind of data, composed of static images of Black Pizzutello. In general, static data simulates the images a farmer, or a robot, could collect to perform some agricultural action on specific grape bunches (e.g. quality estimation, disease detection, automatic harvesting). Being able to have good detection and segmentation in both cases is a prerequisite to computing the quality estimation afterwards. This static dataset presented in the publication is small and consists of 134 images of 3000x4000 resolution, collected with the same cellphone camera used for the videos, even if the optics and chip used for video and still images are different, as is often the case with cellphones. This was intentional since it adds a very common source of covariate shift related to the device and capturing mode (motion vs still images). All the images, in this case, have been labelled for detection (bounding boxes), while a small subset has also been labelled for instance segmentation (70 images), using the Innotescus labelling application, described in Section 3.1.2. All these labels are used for validation and testing of the algorithms described in deliverable D4.1 and the related publication [5], where the interested reader can find the details of the method.

3.4.2 Anomaly Detection Data

Tools:

In order to develop a proper training and testing set for the anomaly detection experiments, we developed a small tool to produce square crops of the original data described in previous sections. This tool is implemented in Python and has the following characteristics:

- The application of receives in input cropped (e.g. extracted with YOLO from a bigger image of the vineyard) table grape bunches images
- It generates a number of square crops from the image, taking into account that the crops should contain only table grape pixels (no background). The number of crops and their size is selectable.
- it has a graphical user interface that allows moving the proposed initial crops
- When the proposed crops are acceptable, the application saves them in a user-defined folder together with the cropping box information, for later reference and replicability.

An example of the GUI is shown in Fig. 4. The initial dataset that was sampled using the GUI has been sampled using the images collected with a smartphone (MotoG8) during the 24/09/2021 data collection, using jpg images (not with a grey card for white balance correction), since in this case the accuracy of colour in not relevant for the kind of external defect we are trying to identify and to make the algorithm able to work with generic camera sensors. During the same field trip several pictures of rotten, sick and damaged grape bunches were collected, providing the raw material to crop test examples.

The dataset is composed of two different labels ("good" and "bad"). For each one, 3 sets of crops with different sizes were produced:

- scale_1: 300 x 300 pixels, 150 good samples, 185 bad examples
- scale_1.5: 450 x 450 pixels, 75 good examples, 108 bad examples
- scale_2: 600 x 600 pixels, 115 good examples, 80 bad examples

There is no official training, validation and test split, since, depending on the technique used (anomaly detection or classification) the training changes. More details on the use of this data are given in Section 4.1. Examples of these patches are shown in Fig. 5.

This project is funded by the European Union G.A No 101016906

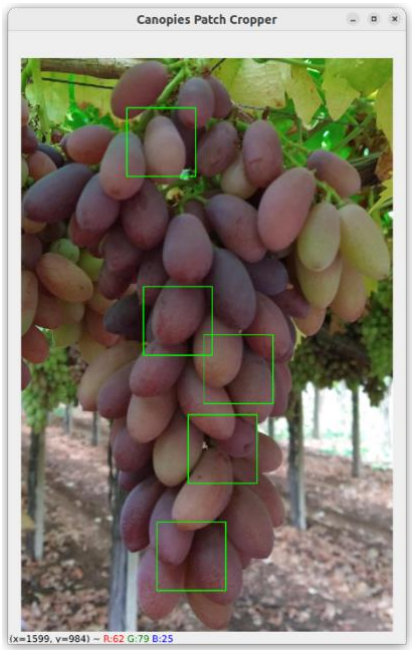

Figure 4 The GUI of the anomaly data acquisition generates random crops that can be moved as needed

Figure 5 Three examples of good patches (up) and three of bad patches (down)

3.4.3 Color and Brix Data

While brix and colour are two different quality characteristics of table grapes, one internal, the other external, our estimation process based on the use of RGB cameras is bound to try to find correlations between them.

From an agronomic perspective, brix degree is loosely related to colour development, meaning that there are a number of factors influencing the development of both, some of which are in common, some not. This means that depending on the weather, on the health of the vine tree, on water availability, on sun exposure, and other factors, the colour could develop more than brix, or vice-versa.

Given that, the harvesting procedure that is commonly followed in table grape production is still based mostly on colour. In brief, an agronomist would sample some bunches both for colour and brix degree (using a refractometer or similar instrument), and, if the characteristics of both colour and brix are good enough, would give instructions to the farmers on which grape bunches to harvest or not.

Knowing this, we hypothesised that the correlation between colour and brix degree is still enough to be able to estimate both using RGB images. We then consider the mismatch due to unknown latent variables as a form of noise in the measurements, that the learning algorithms will have to learn to remove. This approach is also based on the requirements expressed in WP2, and detailed in D2.3, where the desired accuracy for colour (around 80% accuracy in correctly harvested bunches) is a more strict requirement than the error in brix° estimation (that should be less than 5°, a very large relative error).

In consideration of the primacy of colour in this respect, in the early analysis of the problem and after the first data acquisition session (24 September 2021), we also considered how standard camera devices modify image colour in presenting and saving to memory the images. One of the early questions that arose was if the colour constancy between different images, illumination conditions, and camera devices could heavily impact the results of colour-based estimation algorithms and, eventually, how to mitigate this. Further considerations in this respect are given in Section 5.2, but here we say that to ensure being able to answer the question, we collected a considerable amount of images and video sequences both in jpg and raw format using a neutral grey card to be able to correct the white balance and be able to ensure colour constancy between different camera devices and illumination conditions.

For all the aforementioned reasons, the data procedure for brix and colour was mostly the same. In Section 3.4.3-i we give a description of the tools and procedures used to collect and normalise the data, and in Section 3.4.3-ii we describe how the data was organised for the experiments that are finally described in Section 5.2.

3.4.3.1 Tools

Refractometer: The brix° ground truth is one of the most challenging data to collect. The accurate measurement of brix° would require laboratory equipment, but in-field measurements are possible. Agronomists use refractometers, analog or digital, to sample the brix° and make harvesting decisions.

Particularly valuable is the contact refractometer, which does not require plucking and squeezing the berry. For this reason, we acquired an Atago Hikari PAL2, shown in Fig. 6, and verified that had a comparable accuracy to the destructive version, that was in the availability of one of our partners.

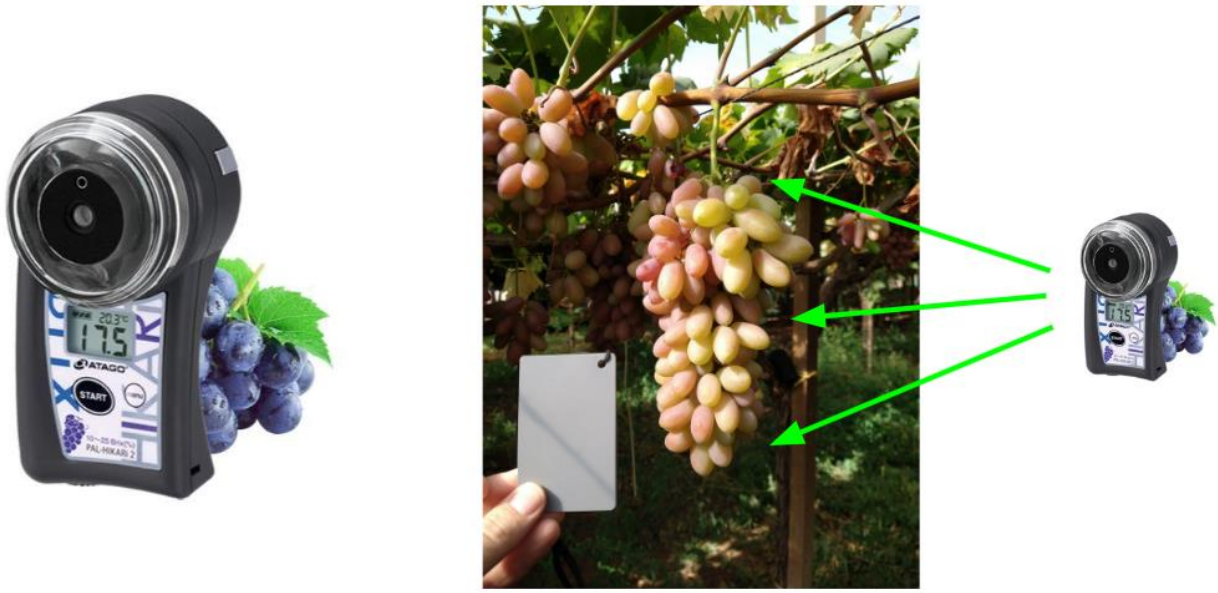

Figure 6 Left: Atago Hikari PAL2; Right: ideal depiction of the sampling process.

The common practice for brix° field measurements is to sample the value from the top, middle and lower sections of the bunch, and then average the value. We collected the ground truth with this threefold procedure, keeping all the measurements in the log, but using the average brix° as a label for the grape bunch images. The procedure is depicted in Fig. 6.

Colour charts: The colour label is based on a reference scale created by AGRIMESSINA. This reflects the quality levels they uses in categorising the products and gives their providers a reference on what quality levels they take into consideration. We have three reference charts for white, black and red table grapes, each one with 6 quality levels, where 1 is the best and 6 is the worst. In Fig. 7 the colour chart for red grapes is shown, together with the threshold we considered for harvestable quality level.

In recent years, due to the high temperatures, the Pizzutello Nero has not reached an ideal black colour (only 15% of the production); the predominant colour is light pink or dark pink, and part of the production does not reach an acceptable colour (it has white berries) and cannot be commercialised. For this reason, we used the red chart as a reference in the labelling process.

As mentioned above, we validated our labelling process by comparing it to the AGRIMESSINA one on a small batch of images. The details of this test are given in Section 5.2.2.

Grey card: Grey, white and black cards are used for the calibration of optical sensors. Grey cards allow for white point balancing by imposing that the pictured grey is the mid-tone grey. White and black have not been used in this data collection.

White balancing tool: White balancing plays a crucial role in ensuring the accuracy and reliability of training neural networks that use colour images as input. When training a neural network for tasks such as image classification, it is essential to have consistent and correct colour representations across the dataset. For this reason, we developed a tool for white-balancing raw images that were collected with a grey card in view.

A raw RGB image can be thought of as a measurement of the scene radiance from a camera sensor within the visible range of the spectrum. The grey card used in our experiments serves as a neutral reference material, and its colour should have equal contributions from red, green, and blue. Usually, in practice, this is not the case, and the colour of the grey card corresponds to the colour of the illuminant L=[LR, LG, LB]. Taking this into consideration, we can recover the pixel values of the image as if it were illuminated by a perfectly white light source by applying a diagonal matrix

With this into consideration, we can recover the pixel values of the image as if illuminated by a perfectly white light source by applying a diagonal matrix MWB=[LG/LR, 1, LG/LB]:

IWB=IRAW MWB , where IWB is the resulting white-balanced image.

The implemented application can thus eliminate any incorrect colour cast on the images as a result of incorrect white balancing during the acquisition of the photo. This enables a more accurate estimation of the colour quality and brix degree of the grape bunches present in the images.

Specifically, the user needs to select an area inside the grey card, which serves as a reference for estimating the colour of the illuminant by averaging all the pixels inside the selected area. The tool then uses the extracted ground truth illuminant to generate a white-balanced linear image using Rawpy[6] and then PIL to generate the final jpg image. This pipeline guarantees that the colours in the resulting image appear natural, regardless of the original lighting conditions during image capture.

3.4.3.2 Brix and colour dataset organisation

We collected the Brix and colour data over the 2021 and 2022 summer seasons. The target variety for the project is Pizzutello Nero, but we were able to also collect data for Red Globe and Black Magic varieties since they have different ripening periods. We used these two extra varieties mainly to test and perfect the acquisition procedure, while they were not used for designing any estimator. For these reasons, no further processing has been done of the raw data collected for these two variants and in the following, only the data used for the experiments and the design of the system is described. All the data collected is an asset of the project and will be made available to the general public in a similar way to what will be done for Pizzutello Nero.

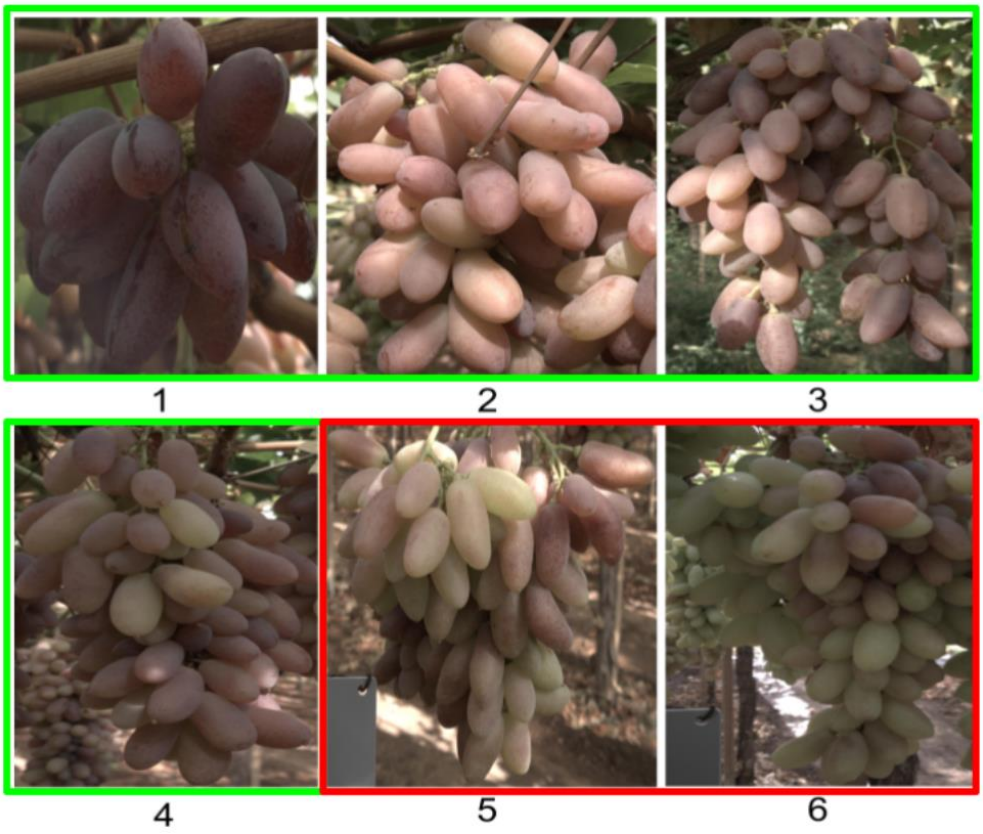

Figure 7 An example of AGRIMESSINA colour charts to classify the colour quality parameters for commercial table grape. In green and red, the threshold for harvesting decision is shown,

The data was collected in different visits, roughly separated by a week or two weeks of time, to be able to sample different colour and brix° stages. The field visit dates are the first data categorization. During each visit, a number of camera devices were used to collect images of the same grape bunches, sometimes in group pictures, and sometimes in close-up pictures. During the 2021 field visits the camera used were a smartphone (MotoG8) and a Reflex digital camera. During the 2022 field visits, we dropped the use of the Reflex camera, using in its place the d435i camera.

Image collection followed this procedure:

1. A vineyard row was selected, based on the number of bunches present

2. Moving along the row, sequentially a grape bunch or group of grape bunches were considered
3. For each grape bunch, the following steps were followed:
    a) picture with the smartphone was taken, both in jpeg and raw format
    b) a picture with the Reflex/d435i from a similar point of view was taken
    c) Each bunch considered in the picture that is not too far or occluded was sampled for brix degree using a refractometer and the three-way sampling technique previously described
    d) All the required notes to be able to correctly associate the bunch image with the brix label were annotated for later post-processing
4. The aforementioned steps were repeated until a predefined number of bunches was sampled for each field visit (usually around 50 samples)

We stress how this procedure was cumbersome and time-consuming, limiting the number of samples that could be collected each day.

Except for the very first visit, in all the sessions both jpeg and raw images were collected with the grey card. Given the complexity of the collection procedure, some errors were made, and thus some images were collected without the grey card. All these characteristics (field visit data, sensor, format, presence of grey card) were considered to make a classification of the data that is summarised in Table 2. Thanks to this analysis, we defined the atomic splits that compose the experimental splits, naming them A, B, C, D, E, F, G, H and I.

To help the reader navigate Table 2, the A, B, C and D splits are related to the data collected in 2021. This data is the one with both smartphone and Reflex images. Split A is the one without grey card and raw images, for this reason, the images collected during that field visit, while abundant, were mainly used for other purposes (e.g. for anomaly detection, Section 3.4.2), or for comparison with white balanced images.

Splits E, F, G, H, and I refer to the data collected in 2022 with the smartphone and the d435i. Splits E and G were collected without grey card reference, and were not used in estimator training and validation and are marked in orange in the table.

The column “Expert GT” refers to the possibility to have a labelling performed by an agronomist using the white balanced images (from any of the sensors in each day), under the hypothesis that white balancing was needed to get a coherent colour labelling. About the verification of this hypothesis, an

experiment was carried out and is described in Section 5.2.2. The column “Color GT” is the colour labelling we produced, following the AGRIMESSINA guideline described in Section 3.4.3-i.

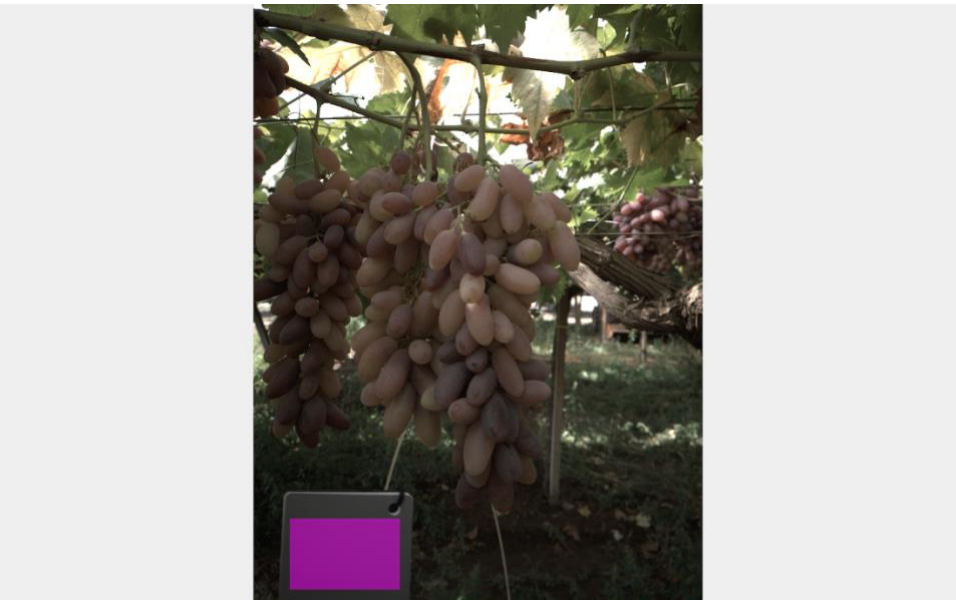

Figure 7 A screenshot of the GUI used to compute white balance using the grey card.

From these atomic splits, the actual dataset splits were composed, as described in Table 3 and Table 4. Table 3 refers to the Basic Splits, i.e. splits that were made according to year, image format and white balancing. These splits were used for the early experiments to test the different hypotheses we had. Table 4 instead describes the Mixed Splits, that were assembled to test covariate shift and to train and validate the brix and colour estimators on the greatest amount of data possible. The details of the experiments are given in Section 5.2.5.

Table 2: This table shows the atomic splits, i.e. the minimum splits that could be defined according to the considered characteristics: i) field visit date; ii) camera device; iii) image format; iv) availability of controllable white balance correction;

Dataset Id	link to folder	year	smarphone				reflex		d435			num images	num samples	Expert GT	Color GT
			type	jpg	raw	card	raw	card	RGB	IR	card				
A	field trip 2021.09.24	2021	motog8	yes	no	no	no	-	no	no	-	100	100	no	no
B	field trip 2021.09.30 A	2021	motog8	no	yes	yes	yes	yes	no	no	-	38	38	possible	yes
C	field trip 2021.09.30 B			yes	yes	yes	yes	yes	no	no	-	62	62	possible	yes
D	field trip 2021.10.07	2021	motog8	yes	yes	yes	yes	yes	no	no	-	50	50	possible	yes
E	field trip 2022.08.08 A	2022	realme	yes	yes	no	no	-	yes	no	no	1	2	no	no
F	field trip 2022.08.08 B	2022	realme	yes	yes	yes	no	-	yes	no	yes	6	22	possible	yes
G	field visit 2022.08.22 A	2022	realme	yes	yes	no	no	-	yes	no	yes	1	2	no	no
H	field visit 2022.08.22 B	2022	realme	yes	yes	yes	no	-	yes	no	yes	4	19	possible	yes
I	field trip 2022.09.01	2022	motog8	yes	yes	yes	no	-	yes	yes	yes	8	51	possible	yes

Table 3: the splits defined in this table are aggregations of the atomic splits, mainly according to device type and white balance criteria. These splits have been used initially to test the covariate shift and the white balance influence on the estimation.

Basic Splits		
name	**components**	**description**
2021PhoneJPG	A+C+D	Contains all the jpg images collected with smartphone in 2021, but without white balance.From A, C and D splits.
2021PhoneJPG_reduced e 2021PhoneJPG_WB_reduced	C+D	Contain respectively, all the jpg and all the white balanced images from 2021 that have a corresponding white balanced image both taken with the smartphone. Corresponds to C and D splits.
2021PhoneJPG_WB	B+C+D	Contains all the jpg images collected with smartphone in 2021, from B, C, D splits
2021ReflexJPG_WB	B+C+D	Contains all the jpg images collected with smartphone in 2021 from B, C and D splits
2022PhoneJPG_WB	F+H+I	Contains all the jpg images collected with smartphone in 2022 from F, H and I splits
2022d435iAutoWB	F+H+I	Contains all the auto white balanced jpg images collected with d435i in 2022, from F, H and I splits

This project is funded by the European Union G.A No 101016906

Table 4: These splits put together the images of the same grape bunches taken with different devices. The images collected with d435i are not in RAW format, and thus cannot be corrected with white balancing techniques, but we found that the post-processing feature of the camera was weaker than the one of the smartphone cameras. We used these splits to test the possibility of mixing both white-balanced images and d435i images to have a stronger estimator.

Mixed Splits			
name	**train set components**	**test set components**	**description**
Season21ToSeason22Phone	A+C+D	F+H+I	All jpg images from phone.
Season21ToSeason22Phone_WB	B+C+D	F+H+I	All jpg images taken with phone that have been white balanced.
Season21ToSeason22All_WB	B+C+D	F+H+I	All phone and reflex images white balanced.
All_Std	A+C+D+F+H+I		All phone images not white balanced, plus all d435i images
All_WB	B+C+D+F+H+I		All phone and reflex images white balanced, plus all d435i images
MixedSeason	A+C+D+H+I	F	All jpg images, keeping for test only the ones the most uncertain brix value (around 16)
MixedSeason_WB	B+C+D+H+I	F	All white balanced images , keeping for test only the most uncertain brix value (around 16)

4 Quality Estimators

4.1 Anomaly Detection Module

The Deep Learning-Based Pipeline for Anomaly Classification in Grape Disease Detection and Localization was designed to accurately identify and classify grapes into two classes (Healthy, Anomaly) through data-driven techniques. By leveraging deep learning-based classification and global average pooling, this pipeline offers an efficient and reliable solution for detecting and localising anomalies in grape plants.

It is important to collect examples of defects in order to train the classifier. While there is an abundance of healthy grape bunches, sick, damaged, rotten, and desiccated are less common. However, since the experimental field is not intended for high-quality table grape production, many grape bunches showed important defects. This diverse dataset serves as the foundation for training a deep-learning model to recognize and classify defects.

A CNN backbone is employed to extract intricate features and patterns from the input images. The trained model undergoes rigorous training and validation processes to optimise its performance and enhance its ability to differentiate between healthy and sick grape plants.

To further refine the anomaly classification process, global average pooling is added to our classification model as a feature aggregation technique. Global average pooling enables the model to summarise the spatial information within an image and reduce the dimensionality of the extracted features. This process not only helps in achieving faster inference times but also enhances the model's ability to generalise well to unseen grape disease instances.

In other words, a GAP not only provides information about the object present in an image but also indicates its precise location within the image effortlessly. This localization is represented by a heat map, known as a class activation map, which utilises a colour-coding system to highlight the regions that hold significant relevance for the GAP-CNN in carrying out the task of object identification.

The process of global average pooling involves calculating the average value of each unit's feature map in the last convolutional layer. These values are then combined using weights to generate the final output. Similarly, we generate class activation maps by taking a weighted sum of the feature maps in the last convolutional layer.

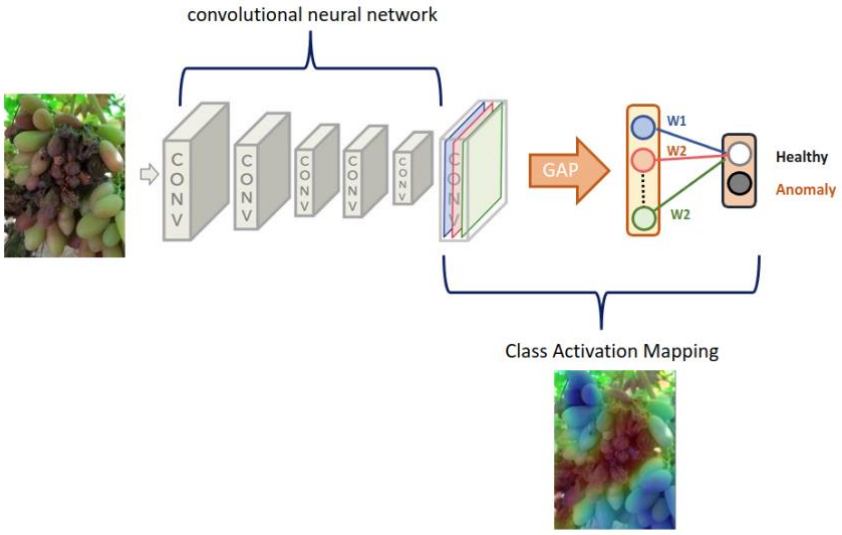

Figure 8 Anomaly Classifier architecture

4.2 Color and Brix Estimation Module

The initial experimentation on brix and colour estimation started in parallel with different approaches, and later in the development of the task work we decided to put together the results in a common estimation framework. In the following subsections, we describe the different approaches and the final architecture used for the QEM.

Colour estimation is a multi-class classification problem. The classes are the colour level described in the AGRIMESSINA chart mentioned in Section 3.4.3-i. The labelling of the colour data has some subjective variability, and an experiment to verify that our labelling was consistent with the one made by an agronomist has been performed and is described in Section 5.2.2.

Brix° estimation is a regression problem, where the measured brix value can range between a minimum of 9° to a maximum of 25°. Clearly, being the problem framed as an image-based estimation, the natural computational tool of choice are Convolutional Neural Networks. We tested a number of backbone architectures on the available data to get a deeper understanding of the problem, and are described in the following subsections. The actual experimental results are presented and commented on in Section 5.

Since the QEM will run on embedded hardware, while being an experimental setup, we also made some early considerations on computational costs. For this reason, we considered using classic histogram-based features both as a baseline for what was possible, and to have a lightweight GPU-free alternative to the CNN-based estimators.

4.2.1 Hand-crafted features for brix estimation

Feature design:

The histogram-based features have been constructed starting from a very basic component, the colour histogram, building up towards more complex features, while experimenting. The main

intuition is that human experts classify colour according to the intensity of colour and its distribution. The more saturated the red colour, for example, the better. The more distributed across the bunch and each berry, the better. Is difficult to define a closed-form feature extractor for this generic observation, but it is reasonable to consider the histogram of the RGB content as the basic building block of this feature.

The input image for the feature extractor is considered to be the cropped bunch extracted by a detector. Since the naive application of the RGB histogram across the whole image would lose all the spatial information, we divided the grape image into cells, and computed the RGB histogram of each cell, concatenating the results. We noticed that the shape of the grape bunches in the cropped images often has background pixels in the four-corner area, mostly in the lower left and right corners. To reduce the feature noise and feature dimension, we decided to experiment with dropping these blocks, according to a cross-shaped pattern. We defined two possible cross shapes, called “fat” and “thin”, and compared the features without any cell drop. Fig. 9 gives an idea of the image division in cells.

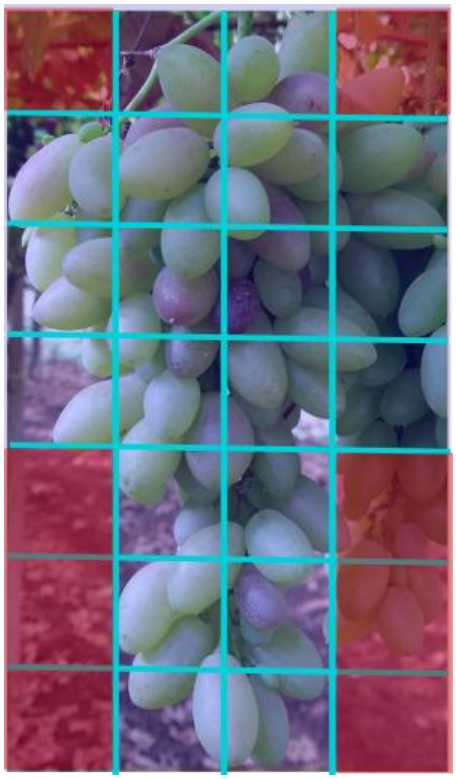

Figure 9 This image gives an example of the binning used for feature extraction. The red areas are cell that are dropped, since with high probability will contain mostly background pixels. The number of cells in the width and height directions defines the cell size (that in general is rectangular) and the rgb histogram is computed as the average of the three channels in each cell.

In summary, the hyper-parameters that were considered while designing the features are:

- Image starting size (width, height)
- Number of cells in both the width and height directions
- Drop of some cells according to a “cross pattern”
- Use of HSV instead of RGB channels

In addition to RGB histograms, to increase the feature capacity to capture the distribution of colour, Histogram of Oriented Gradients (HOG) have been added to each bin, as an optional parameter. However, the algorithm was slightly modified, since the standard HOG extracts the oriented gradients from the grayscale image, while we need to keep the colours and to know the gradients of each channel, particularly red and green.

Estimation algorithms

To keep the estimator computation CPU based, we used as candidate estimators Ridge Regression, Support Vector Machines (SVMs) and AdaBoost. In particular Ridge Regression was used for the early exploration of the feature performances, while the SVM and AdaBoost were tested because of the No Free Lunch theorem [1].

The results of the experiments and of the hyper-parameters search are given in Section 5.2.4.

4.2.2 Early colour classification with MobileNetV3

The problem we wanted to address in this first iteration of the module was the binary classification of the grapes into good and bad colours. This would provide the robot with a simple binary decision regarding this quality indicator. Taking into account the colour quality levels described in Section 3.4.3-i, we assigned positive labels to the colours going from 1 to 4, and negative labels to colours 5 and 6, indicating whether the grape bunch has an adequate colour for harvest.

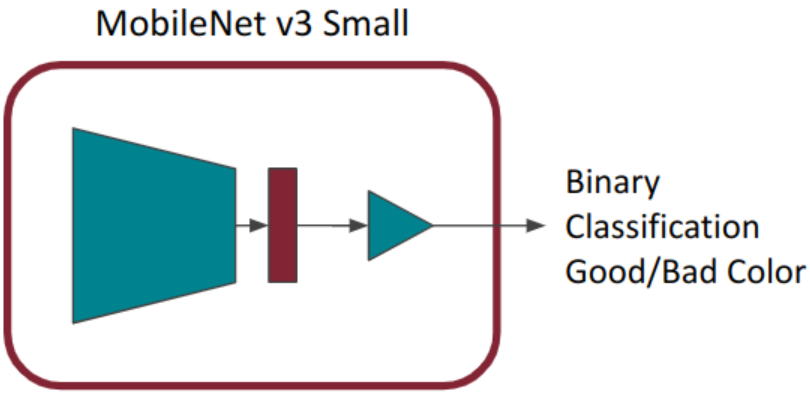

Figure 10 Architecture for the baseline color harvesting classifier

The pipeline we implemented first goes through data preprocessing to ensure that the grape images are in a suitable format and size for input into the MobileNet V3 Small model. In this case, as for the above-mentioned brix estimation model, the input for the model is assumed to be the cropped bunch extracted by the detector.

The model's choice is dictated both by the scale of data collectable for the problem and by the necessity to be run on an embedded, battery-run GPU. The model is finetuned starting from weights pre-trained on ImageNet. This step reduces the need for large training datasets and accelerates the training procedure. During training, we employed a weighted binary cross-entropy loss. By assigning appropriate weights we aimed to mitigate the potential class imbalance and encourage the model to focus on accurately classifying the colours.

Although the model achieved good performance on the validation set, it showed some performance degradation when tested in the field. This was primarily caused by the model's insufficient generalisation to various backgrounds and lighting conditions. The full validation description is given in Section 5.2.3.

4.2.3 Representation learning with multi-task architecture

Following the intuition that brix° has still to be learned from visual colour cues, we decided to opt for a multi-task approach to estimation, using neural networks. Multi-task with backpropagation is easy to set up, while it is very difficult to do with traditional optimization algorithms, such as SVM since they are more of an academic niche when compared to the technical maturity of NN solutions. For

this reason, the multi-task approach was designed from scratch, without using the histogram features, even though it is based on the intuitions gained through the experiments involving them.

Network architecture

Representation learning for multi-task learning requires setting up architectures that can extract some structure from the images, while at the same time learning how to classify and regress the target tasks. In our case, the choice fell on having a backbone auto-encoder as the basic feature learner, while two task heads were used for learning the colour and the brix°. The auto-encoder is a UNet with a Resnet18 backbone. The encoded feature layer that is produced by the auto-encoder bottleneck is elaborated by an additional convolutional layer to produce the common feature set that is then used by the two heads. The heads are CNN and the whole architecture is designed in a fully convolutional way to avoid having specific image dimensions in input. Given that, the hypothesis is still that the input comes from a detection step that crops images of table grape bunches, and then rescaled to some common image size, so the choice for using only convolutional layers is mostly to limit the number of parameters in the network.

4.3 Size and Shape Estimation Module

At harvesting time, the size of the grapes is essential because it affects the overall quality of table grape crops, and the harvester can also use it for yield estimation. In particular, two values are important: the size of the whole bunch, measured in centimetres, and the size of the berries, measured in millimetres, which is usually randomly sampled along the vineyard using a calibre.

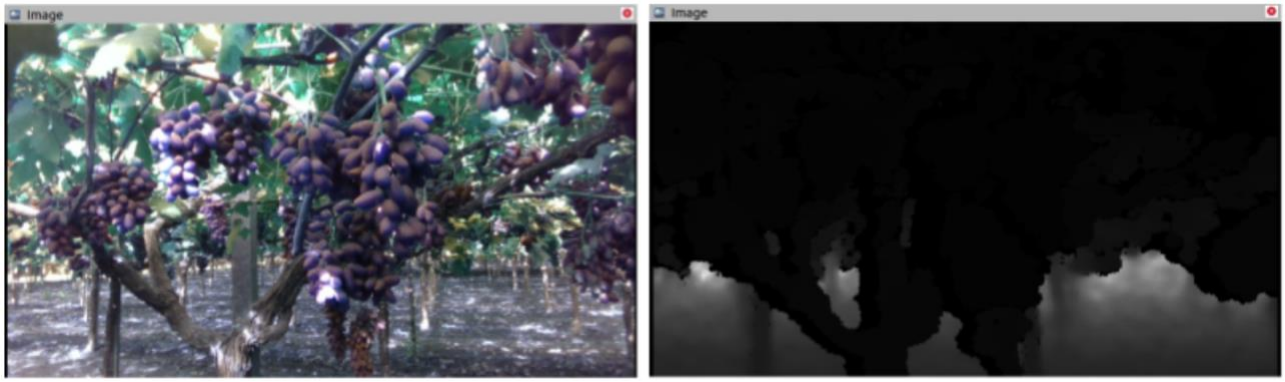

Figure 11 Examples of RGB and Depth output from d435i camera

To measure elements' size, we need metric information of the scene since it is possible to measure the relative distance of two points in space. This kind of information comes from the calibrated depth camera (RealSense d435i) mounted on the wrist of the robotic arm since it provides for each colour frame a depth map of the pixels. Given the aligned colour and depth frames and the calibrated camera, extracting the spatial location of pixels in metres is enough to perform a simple triangulation using the pinhole camera model with the following equations:

$$X = (x - cx)*depth/fx$$

$$Y = (y - cy)*depth/fy$$

$$Z = depth$$

To measure the width and height of the crops, we take the boundaries of the bounding box extracted by the detector, and we measure the height and width by measuring the distances indicated in the picture. The bounding box of the detector consists of 2 points: the left bottom corner (x_b, y_b) and the right top corner (x_t, y_t). Given the coordinates of those points, it is easy to extract the 4 points (a, b, c, and d) to project in 3D to compute the size of the grape bunch.

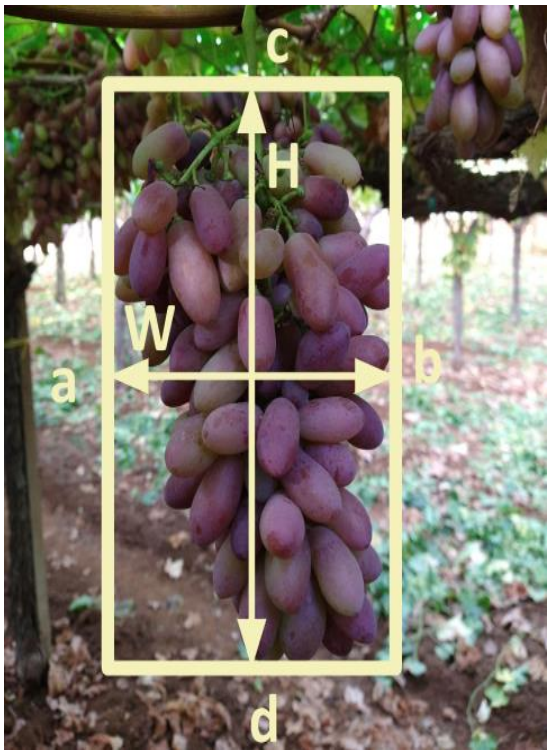

Figure 12 Diagram of the visual measurement of a grape bunch.

a = [xb, yb+(yt-yb)/2]

b = [xt, yb+(yt-yb)/2]

c = [xb+(xt-xb)/2, yt]

d = [xb+(xt-xb)/2, yb]

The RealSense d435i is the depth camera used for this project and uses stereo vision to calculate depth. The stereo vision implementation consists of a left imager, a right imager, and an optional infrared projector. The infrared projector works by projecting a non-visible static IR pattern and is mainly used to improve depth accuracy in scenes with low texture.

However, this sensor is inaccurate in outdoor conditions due to different factors:

- Sensitivity to ambient light: Depth cameras rely on projecting and capturing infrared light patterns to measure depth. In outdoor environments, direct sunlight or other intense light sources can interfere with the infrared signals, causing inaccuracies or complete failure in depth measurement.
- Lack of texture: Depth cameras rely on detecting patterns and textures on surfaces to calculate depth accurately. In outdoor environments, surfaces like vegetation, water, or reflective materials may lack the necessary texture or have complex patterns that confuse the depth-sensing algorithm. This can result in inaccurate depth measurements or missing depth information altogether.
- Interference and noise: Outdoor environments often contain various elements that can interfere with the depth-sensing process. These can include dust, fog, rain, or other particles in the air, which can scatter or absorb infrared signals, leading to reduced accuracy or noisy depth data.

Given those issues, sampling the depth on the edges of the bounding box is unreliable. Therefore the depth is sampled in the middle of the bounding box, where we are sure that a reasonable value is provided, and then we proceed to use that depth value to project back the 4 borders in 3D.

To measure the size of the berries, we proceeded in the same way, but instead of using the detector of the grapes, we trained another detector to extract only the berries. This detector uses the same model as the detector of grape bunches, which is YoloV5, but instead of taking the whole image as input, it only takes the cropped image of the grapes detected by the previous detector.

To train the berries detector, we took a total of 28 images of grape close-up and manually annotated the single berries. The number of images is not very high, but those are sufficient to train the YOLO detector well enough because many instances are present in each frame.

4.4 Detection and Tracking

The grape detector is the first part of the quality estimation pipeline since all the other modules take its prediction as input. In single-shot object detection, YOLO represents the state-of-the-art; in particular, we choose the YOLOv5[7] version, which Ultralytics provides because of its real-time performance and ease of deployment. Moreover, since all the algorithms have to run in parallel on the same hardware, we choose the 'S' version of YOLOv5 since it is the one that, for our application, provides the best accuracy while working in real-time and consuming a tiny portion of GPU memory.

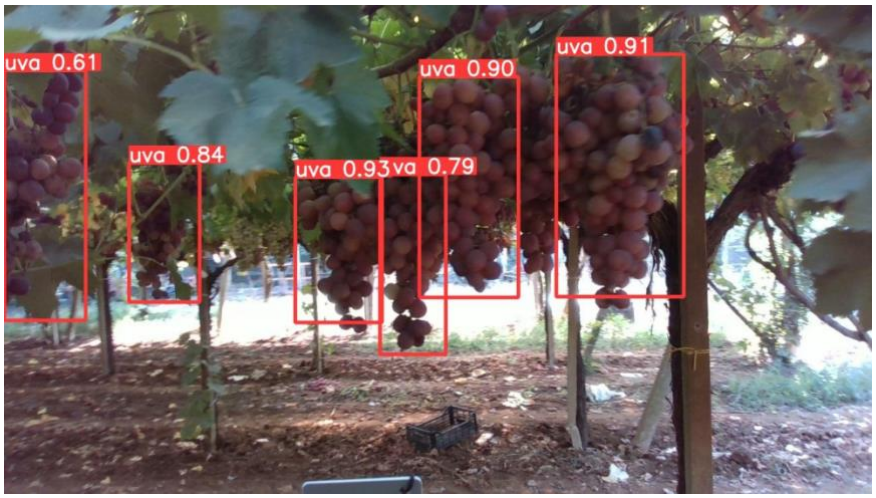

Figure 13 Example of YOLOv5 detector operation

One of the problems we faced was the lack of labelled data and the fact that object detection algorithms are data-hungry, meaning that they require a lot of labelled data to be appropriately trained. However, to produce a large quantity of labelled data, it takes a lot of time and a lot of human resources to collect and annotate all the data. To overcome this challenge we developed a pseudo-labelling tool[5] that has been described in deliverable D4.1 since it is also a foundational tool for task T4.1. The detector allows the extraction of information about the location of grape bunches in single frames. This information is undoubtedly useful for estimating the quality, but it would be more

Figure 14 Example of berry detection.

beneficial to integrate the quality information obtained across multiple frames. This is because a grape bunch may seem good on one side but seen from behind, it may be entirely not mature. For this reason, we exploited the multiple object tracking paradigm (MOT) that allows us to identify and

locate objects of interest in each frame and associate them across frames to keep track of their movements over time.

The challenges to this problem are occlusions by other crops or by environmental elements like branches and poles, motion blur deriving from high-velocity motion, and the appearance of each grape bunch being very similar. There are different approaches to the tracking problem. Still, the tracking-by-detection paradigm is one of the most commonly used, allowing for real-time performances while exploiting the strong representational power of modern detectors. The pipeline implemented by us consists in taking the detection at the current frame to associate with the tracks from previous frames, which are instantiated using a Kalman Filter formulation. The association is based on an Intersection over Union (IoU) cost matrix between the current detection and the tracks propagated, which is solved using the Hungarian Algorithm. This combinatorial optimization algorithm solves the assignment problem in polynomial time and represents the state-of-the-art.

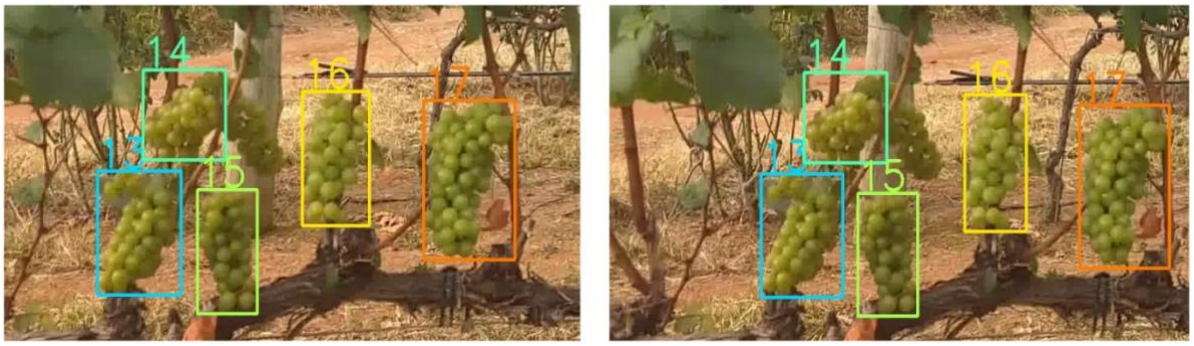

Figure 15 Example of tracker operation.

This project is funded by the European Union G.A No 101016906

5 Experiments and Discussion

5.1 Anomaly detection validation

The first step in evaluating our approach for anomaly classification is to prepare a test set of labelled images, consisting of 150 images depicting both healthy and anomalous cases. This test set is exclusively used for testing purposes.

Next, we utilise a state-of-the-art method in grape anomaly detection, which is built on an autoencoder[8]. The approach follows a reconstruction-based strategy. If an all-convolutional autoencoder (CAE) learns a specific stochastic mapping and a sample is presented to the trained network that does not belong to the trained distribution, it exhibits a higher loss compared to that caused by a training sample. This increase in loss can be interpreted as an anomaly.

For the baseline architecture, we employ a symmetric all-CAE comprising four convolutional encoder and decoder layers. Both weights and biases are subject to L1 regularisation.

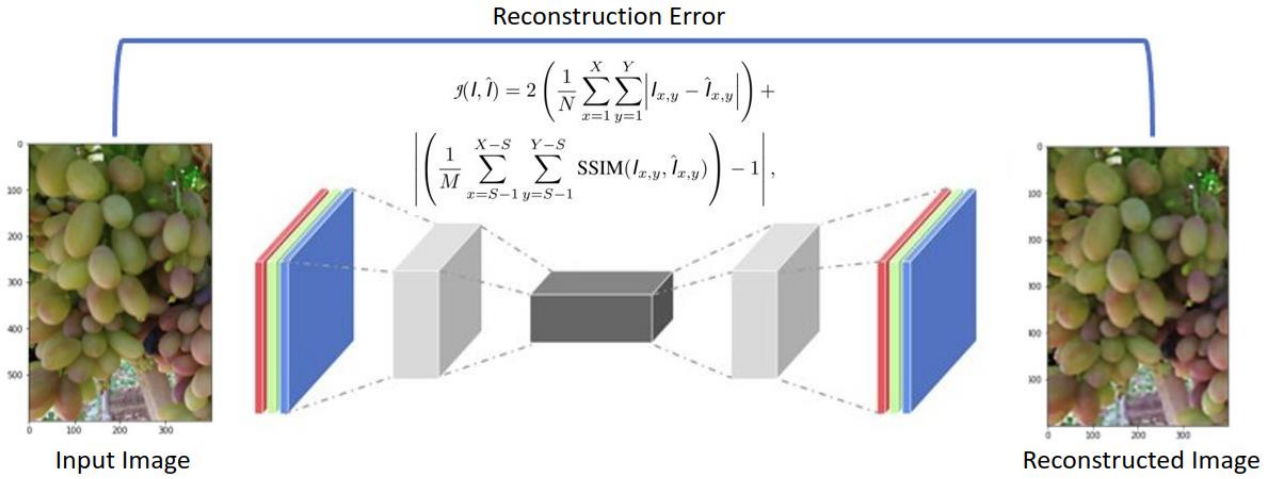

Figure 16 Architecture of the anomaly detection auto-encoder baseline

In evaluating the performance of two classification methods, it is crucial to consider appropriate metrics that accurately reflect their effectiveness in detecting anomalies. For this purpose, we have chosen to focus on two key evaluation measures: recall and precision.

Recall, also known as sensitivity or true positive rate, measures the ability of a classifier to correctly identify positive instances, specifically anomalies in our case. A higher recall indicates that the classifier has a greater capacity to capture anomalies, minimising the risk of false negatives or missed detections.

Precision, on the other hand, assesses the accuracy of the classifier in labelling instances as positive, in our case as anomalies. It quantifies the proportion of instances classified as anomalies that are indeed true anomalies. A higher precision suggests that the classifier produces fewer false positives, reducing the chances of misclassifying normal instances as anomalies.

By considering both recall and precision, we gain a comprehensive understanding of the performance of the two methods. While recall emphasises the ability to detect anomalies effectively, precision highlights the precision of the detected anomalies. By evaluating these two metrics, we can make informed decisions about the trade-off between capturing as many anomalies as possible and ensuring the accuracy of the identified anomalies. Ultimately, selecting recall and precision as our evaluation criteria provides valuable insights into the strengths and weaknesses of the two classification methods, aiding in the determination of their suitability for anomaly detection tasks.

The output of the two classifiers, as indicated by their respective recall and precision values, reveals important insights into their performance for classifying anomalies. The baseline exhibits a recall of 0.69, which means it correctly identifies 69% of the anomalies present in the dataset. Although this suggests that there is room for improvement in terms of capturing all anomalies, it still manages to identify a significant portion of them.

Moreover, The baseline demonstrates a precision of 0.85. This indicates that out of the instances it labels as anomalies, 85% of them are true anomalies. In other words, while there may be some instances that are incorrectly classified as anomalies, a considerable proportion of the identified anomalies are accurate.

On the other hand, the proposed approach demonstrates a notably higher recall of 0.97. This indicates that it is good at identifying anomalies, correctly capturing 97% of the anomalies present in the dataset. With such a high recall value, the proposed approach exhibits a good ability to detect anomalies effectively, minimising the risk of false negatives or missed detections.

The proposed approach also showcases a precision of 0.87, implying that out of the instances it labels as anomalies, 87% of them are indeed true anomalies. This suggests that the proposed approach has a relatively high accuracy in labelling anomalies, indicating that the identified anomalies are likely to be accurate.

In summary, the proposed approach outperforms the baseline in terms of recall, successfully capturing a higher proportion of anomalies. Additionally, the proposed approach demonstrates a slightly higher precision, suggesting that its identified anomalies are more likely to be accurate. These results indicate that the proposed approach is more effective in detecting anomalies compared to the baseline, with a higher ability to capture true positives while maintaining a relatively low rate of false positives.

5.2 Brix and colour estimation validation

The validation of the brix and colour estimation algorithms took different steps. Initially, the problem that had to be solved was to define some early decisions on architecture and data collection. For this reason, we ran preliminary tests together with the experts of AGRIMESSINA to better understand the problem.

Later, after the first data acquisition, we were able to test some baseline algorithms to better understand the complexity of the problem and test how the environment could impact the quality of the estimates.

Finally, after the final data collection, we developed the multi-task approach as the main architecture to enable the robotic platform with both colour and brix estimation capabilities.

5.2.1 Exploratory data analysis:

One of the initial questions of task T4.3 was if to estimate brix° a multi-spectral camera was needed. The acquisition and placement of such a device on the robotic manipulator was a complex challenge, therefore being able to avoid it was of significant interest for the consortium. To understand if the RGB cameras could be enough for the estimation, we designed a simple experiment with the help of AGRIMESSINA agronomists.

We used the jpg smartphone images collected during the first data collection session (24 September 2021) for Pizzutello Nero. The 100 images were given without any other hint to AGRIMESSINA experts, asking them to express a numeric evaluation of the quality (colour and brix guess). The agronomist also gave an evaluation of the lignification of the peduncle and berry shape that were visible in the image, to check if there is a correlation between brix° and other external parameters. We then compared the human prediction with the average brix° measured to have a measure of the correlation between grape bunch appearance and actual internal ripening. An excerpt of the expert evaluation is given in Table 5.

The main objective of the analysis was to find if there is a correlation between the actual internal quality (the binary evaluation on brix degree) and the appearance scores given by the experts, i.e. colour score, lignification score and berry shape score. In Fig. 17 the correlation matrix is shown. The correlation score between the brix evaluation (b_class) and colour evaluation (c_class) is 0.421, while the correlation between b_class and lignification and berry shape scores is much lower. The correlation between colour and brix in this experiment is not high but is still present. However, deepening the analysis, we found that if we raise the brix harvesting threshold to 18°, the precision on harvesting decisions based on colour reaches 1. This is because the correlation between high colour scores and high brix values is higher than the general correlation. This consideration is important if we consider the result in the context of the agronomic operation: the grape bunches that are still good to harvest (brix° >16) but have not been harvested, will be harvested in a future pass of the farmers of the robot. This is an example of Type II (false negative) error. The Type I error in this case is when an unripe bunch is harvested (false positive). This second kind of error is much worse because it cannot be recovered later on.

This exploratory analysis led us to go on with the estimation of brix° only through RGB cameras, simplifying the camera setup of the robotic platform.

Table 5 This is an excerpt of the analysis performed together with AGRIMESSINA agronomists. The experts were asked to give an evaluation of the colour, lignification and berry shaper based only on the image. The colour score, lignification score and berry shape score are based on a 1-10 subjective scale. The Brix eval is the binary evaluation of the minimum harvesting brix° (>16°) based on actual measurements, while Color eval is the binary evaluation of colour quality, based on the subjective evaluation given by the agronomists.

Image name	measure #1	measure #2	measure #3	Mean brix°	Brix eval	colour score	Color eval	Lignification	Berry shape
IMG_0	10,20	9,00	11,80	10,33	NO	1	NO	3	4
IMG_1	11,00	10,00	11,10	10,70	NO	1	NO	8	6
IMG_2	10,30	13,30	11,30	11,63	NO	1	NO	3	8
IMG_3	13,60	14,50	12,20	13,43	NO	1	NO	5	6
IMG_4	10,60	13,80	12,40	12,27	NO	1	NO	4	7
IMG_5	18,10	19,10	13,00	16,73	YES	1	NO	7	6

5.2.2 Color label consistency

Another user study done with the help of AGRIMESSINA agronomists was intended to assess the quality of our internal colour labelling (based on the 1 to 6 AGRIMESSINA scale described in Section 3.4.3-i) and the one given by the experts. In addition, we needed to test if the white-balanced images would be evaluated in a different way by a human expert, in order to understand how much the colour evaluation depends on colour constancy in human assessment.

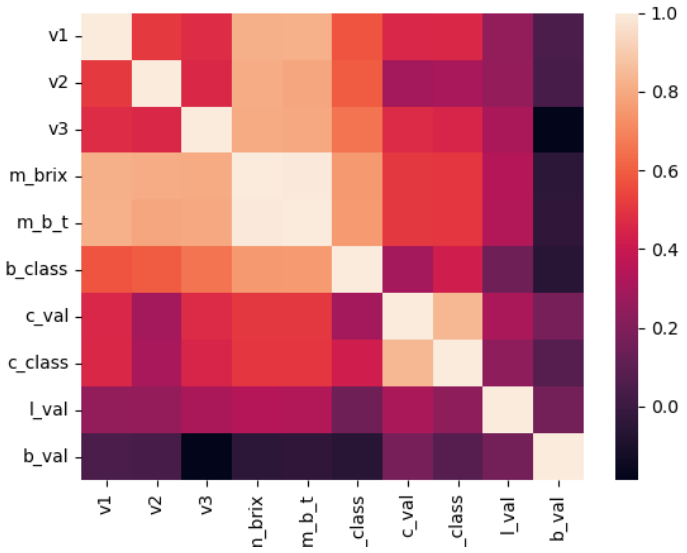

Figure 17 Correlation matrix between the characteristics considered in the expert user study. v1, v2, v3 are the brix measurements; m_brix is the mean brix; m_b_t is the same value rounded; b_class is the binary harvesting evaluation on brix°; c_val is the color score; c_class is the binary harvesting evaluation on color; l_val is the lignification score; b_val is the berry shape evaluatio score;

The study consisted in asking the agronomist to evaluate 50 smartphone images taken from the 10th of October 2021 field trip, both in standard jpg and white balanced version. The agronomists were unaware that the images of the two groups were the same subjects. This time the labelling scale was the AGRIMESSINA scale for red grapes, instead of the arbitrary 1-10 scale used in the previous study.

A short summary of the evaluation is presented in Table 6. The mean difference of colour evaluation between our labelling and the expert one in both colour groups is less than one, with a standard deviation of approximately 0.5. The differences between the evaluations are mostly by one single step in the AGRIMESSINA scale and the noise in the binary harvesting decision is noticeable only when the error is between the fourth and fifth category. This happens in 14 out 50 samples, where in 4 cases with no effect, 9 cases with false negative errors (category 5 instead of 4), 1 case false positive error (category 4 instead of 5). In percentage points, this means that there are 18% Type II errors in the labelling for harvesting purposes, while only 2% Type I errors. Given that type II errors are less severe than Type I, we decided that we could go with our internal colour labelling for the experiments, leaving as a future development the refining of the labels with expert advice.

With respect to the question of how much the white-balanced images had a different evaluation with respect to the standard jpg images, the mean error is small (0.08), but not negligible. This means that even considering the brightness and colour compensation of the human vision system, the noise in the colour labelling is not negligible. More detailed statistical analyses of these issues are left for future research.

Table 6 The first two columns show the mean and standard deviation of the difference in colour evaluation between our labelling and the labelling made by AGRIMESSINA experts. The last two show the difference between expert evaluation in group 1 and group 2 images.

	mean I-E	Std I-E	mean error	error std
Group 1 (Std)	0,54	0,54	0,08	0,59
Group 2 (WB)	0,61	0,53		

5.2.3 Baseline colour estimator validation

The MobileNet-based model, introduced in Section 4.2.2, served as the initial baseline for predicting the colour of grape bunches. In this preliminary model, the desired output was a binary decision indicating the quality of the grape colour (good/bad).

To ensure accurate colour representation, the datasets used for training and validating this model exclusively consisted of white-balanced images. This approach eliminates any influence from an erroneous illuminant estimation by the camera pipeline. For training, we utilised the B and C datasets, detailed in Table 2, while dataset D was reserved for validation purposes.

As described in the estimator description, the labels underwent a processing step to convert the colour levels into a binary decision. Namely, colours from 1 to 4 were categorised as 1

Table 7 Summary of harvesting labels for binary classification based on colour.

Dataset Id	Positive Labels	Negative Labels	Total Count
B & C	8	92	100
D	22	28	50

(representing "good colour"), while colours 5 and 6 were labelled as 0 (for "bad colour"). The summary of the dataset subdivision is shown below:

Given the dataset imbalance, a weighted binary cross-entropy loss was employed. The weight for the positive labels was set to WP=#pos/#neg.

The model was trained for 100 epochs, and the epoch with the best F1-score was chosen as the final model. The performance of the model, when using a threshold of 0.5 on the probability output, is shown in Table 8a and in Fig. 19a. It can be observed that errors are made equally between mistaken positive and negative labels, indicating that the weighted loss effectively handled the unbalanced dataset.

Moreover, by observing the precision-recall curve in Fig. 20, we can see that we can obtain very good precision with a reasonable recall, as shown in Table 8b and in Fig 19b. This means that

Table 8 performance of the binary colour estimator.

Validation Metrics	Threshold 0.5 (a)	Threshold 0.65 (b)
Precision	0.73	0.92
Recall	0.73	0.55
F1-Score	0.73	0.69
Accuracy	0.76	0.78

the robot can autonomously identify the grapes with good colour, leaving to the interaction with the human the borderline cases for which it is not sure.

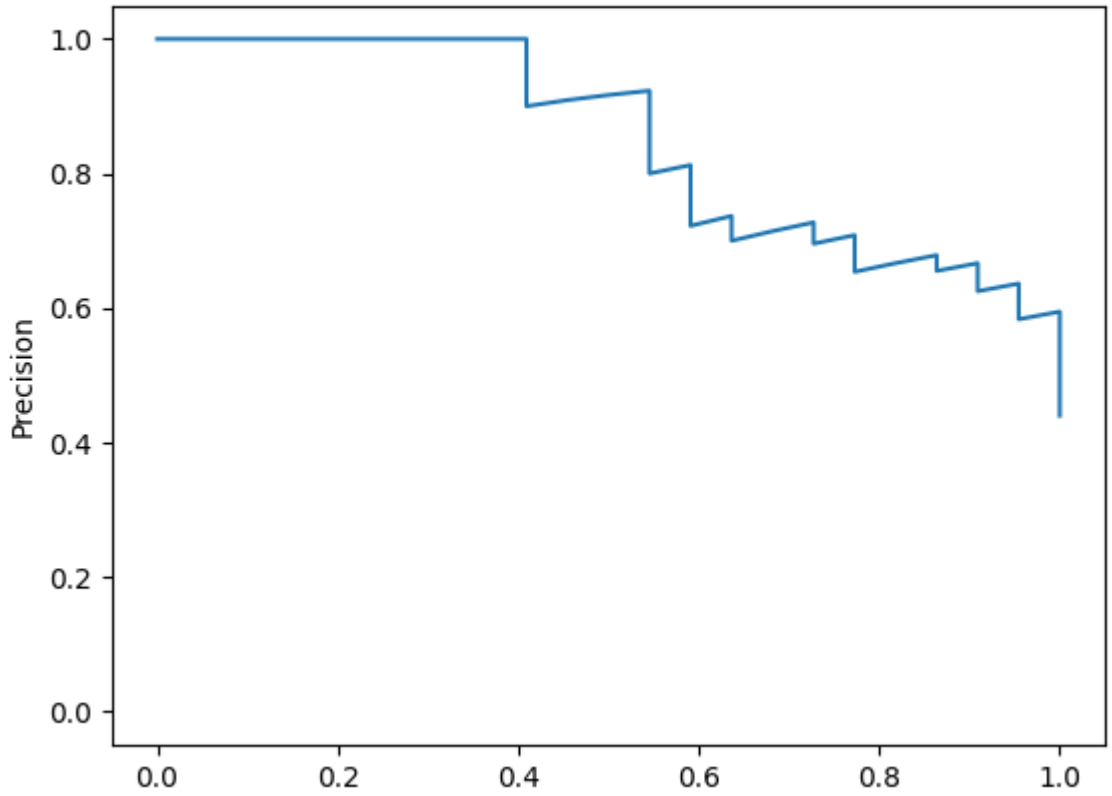

Figure 18 Precision-Recall plot.

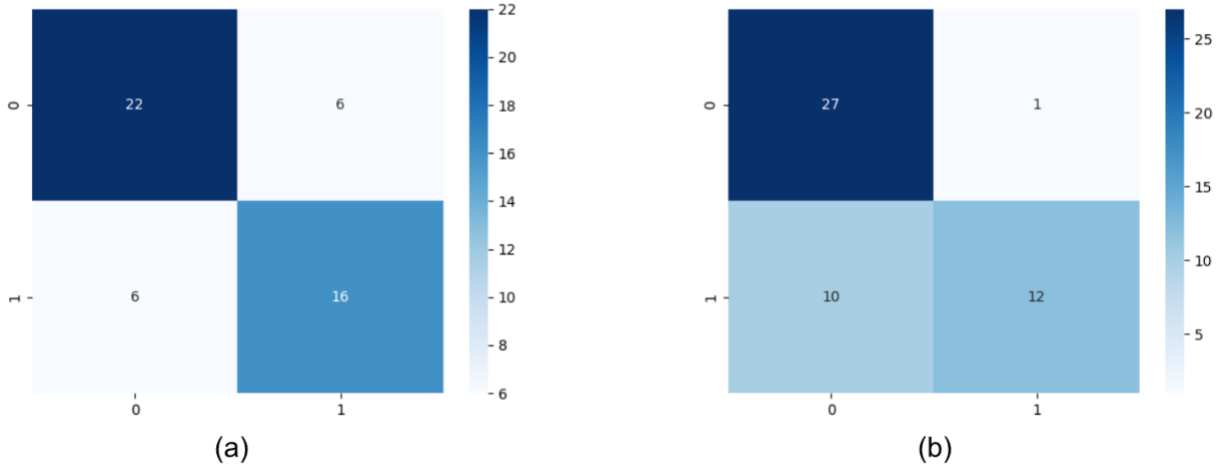

Figure 19 Confusion matrices are shown for the validation set when a threshold of 0.5 (a) or 0.65 (b) are used on the probabilities outputs of the model to obtain the prediction.

In the case of using a threshold of 0.65, the only grape bunch that is mistakenly predicted as having a good colour is the one shown in Fig. 20a. The ground truth for this image is a colour of 5, indicating

that it is not yet ready for harvesting. It's worth noting that being a borderline case, another annotator might label it as a 4, putting it among the positive labels.

Fig. 20b, instead, shows the opposite type of mistake, where a good grape bunch is predicted as a bad one. The reasoning behind this ground truth labelling is that, apart from the desiccated part in the upper-right hand corner, the rest of the grape bunch has a good enough colour for commercialization. A worker in the field would collect the grape bunch and prune off the desiccated green part. However, this type of reasoning or handling precision is not implemented in the current project. Therefore, it is reasonable for the robot to leave these types of grapes on the vine, leaving it to humans to collect them.

Figure 20 Examples of mistakes made by the model with a threshold of 0.65 on its output. In (a) we have a false positive, while in (b) a false negative.

5.2.4 Histogram feature validation

The histogram features described in Section 4.2.1 have been used for different purposes. First and foremost, they are used to model a baseline for brix estimation. Given this baseline, the parameter sweep performed on the initial models was also used to evaluate the effects of white balancing.

Early parameter sweep:

The first set of experiments was done on the highly homogeneous data category at our disposal, in order to reduce the effect of other variables, such as covariate shift due to different devices or years, and to be able to answer specific questions on the estimation problem. The trained model was a Ridge regression, where the L2 regularisation parameter has been called 'alpha'. The training has been performed on different hyperparameters using 5-fold cross-validation. The two dataset splits used for the runs are the 2021PhoneJPG_reduced and 2021PhoneJPG_WB_reduced (See Table 3 for

details). These splits have the properties of being collected the same year, with the same device, and have both the standard jpg and white-balanced (WB) version.

As mentioned before, one of the main questions of this experiment is "Does using WB images improve the performance of the regressor?". The other question is what hyper-parameters to use with both sets. To address these questions, a parameter sweep, with random sampling of the hyper-parameters for a total of 6736 different runs. Without going too much into the details, Fig. 21 shows training and validation errors computed with respect to the alpha parameter in the WB images, while Fig. 23 shows the parameter influence on the validation error. The same plots are shown for standard images in Fig. 22 and 24.

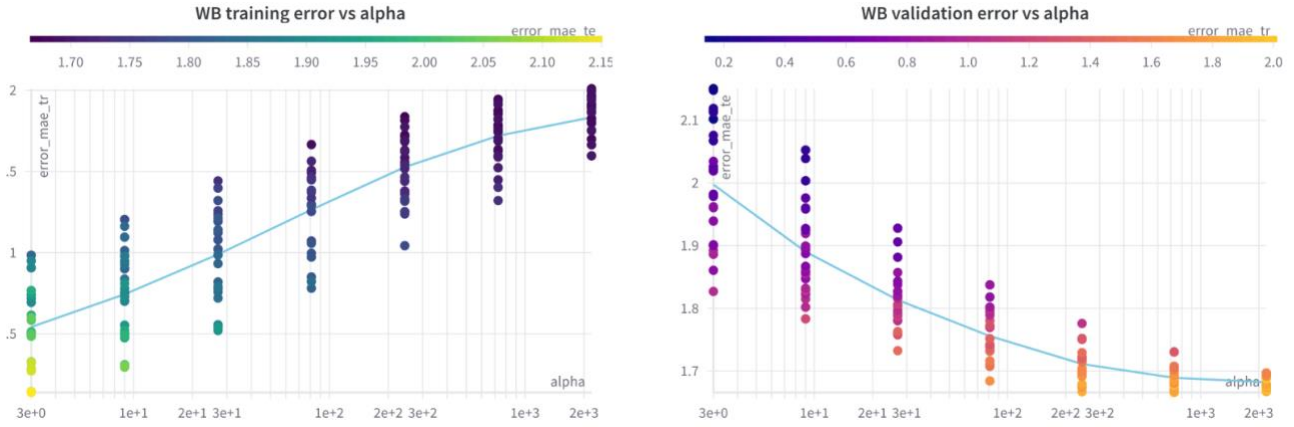

Figure 21 Training and validation curves for WB dataset.

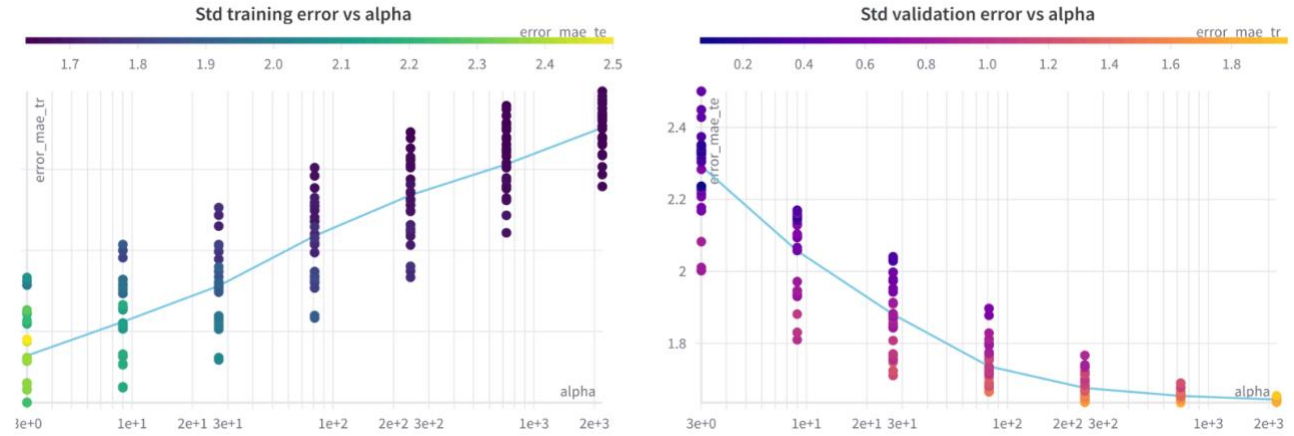

Figure 22 Training and validation curves for standard dataset

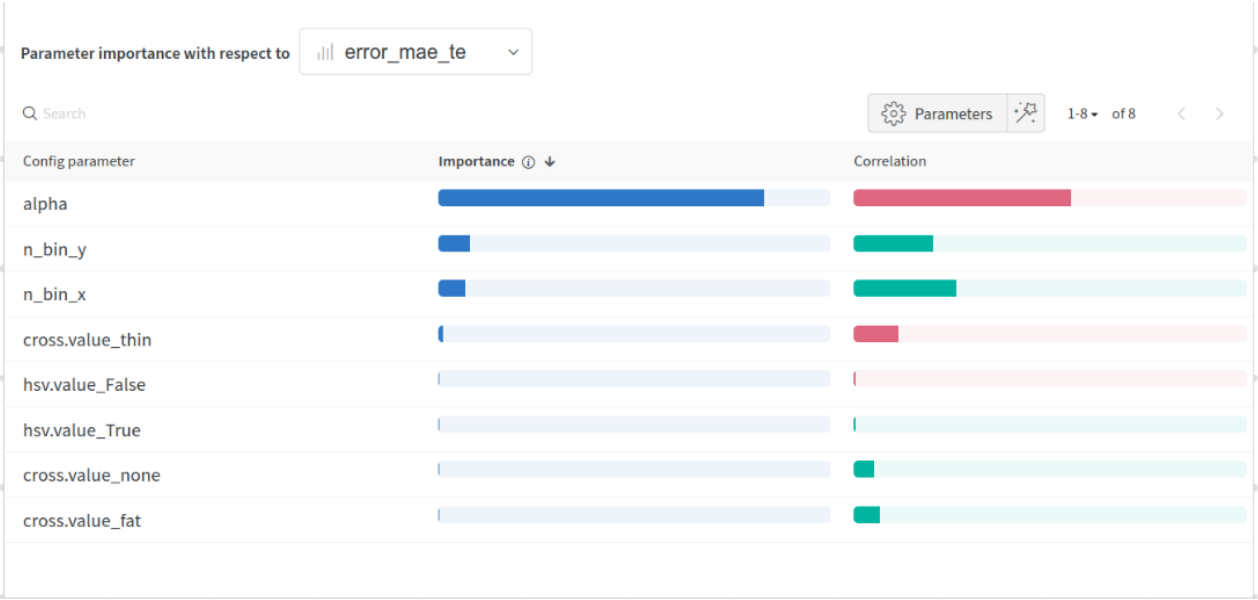

Figure 24 W&B importance view: this diagram shows the relative influence of the hyper-parameters with respect to the relevant validation metric. In this case there is a strong correlation between high values of alpha, together with large bin dimensions in x and y, and with the highest drop of bins (cross="thin"). All these parameters show that the dataset is to small for this features, and it is needed strong regularization to overcome overfitting.

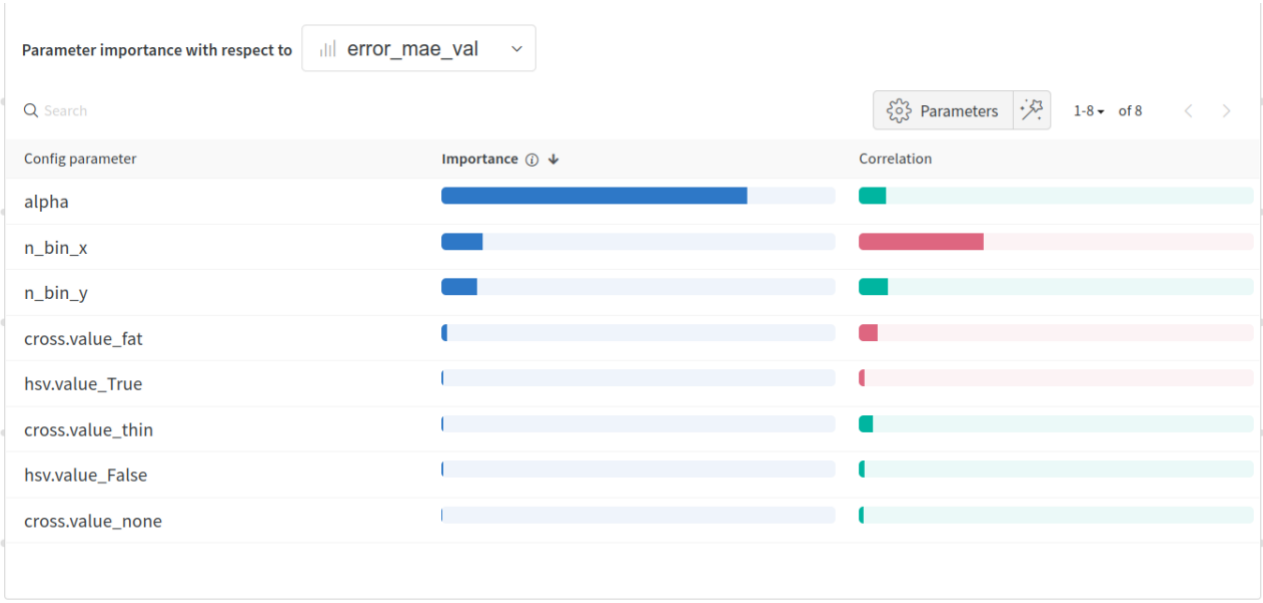

Figure 23 Importance diagram for models trained on standard images. Similar consideration as with the white balanced images could be done. The striking difference is with the x-y bin dimensions, where the x direction should be greater than the y one. In any case, again the feature complexity is to high for the size of this dataset split.

The best runs in both Std and WB reach around 1.665 MAE validation error, with the Std ones showing just a slightly lower training error. The parameter importance shows that the features are too complex for the size of the dataset split chosen since all the parameters improve the performances towards higher regularisation (increasing alpha or diminishing feature size). There is no strong cue on the use of WB images instead of the standard device, however, we realise that from the experiment that real use of white balancing should be tested on cross-seasonal, or cross device training. Training with a split of data that comes from the same distribution, should not be a problem

for a learning algorithm, if the image processing pipeline of standard jpg images is repeatable, as is probably the case.

Finally, while the MAE error seems to achieve the required objective (less than 5°), an absolute evaluation can be done only by looking closely at the basic statistics of the training and testing sets. In this case, the mean brix value for the splits is 17.54 and the standard deviation is 3.07. The actual mean and standard deviation of the validation set depend on the actual samples that fall into it, however, we can compare our estimators with the baseline that predicts the average of the brix° found in the training set and gives an average validation MAE of 2.4. From these considerations we can say that the simple estimator based on histogram features is able to detect signals in this small dataset, validating the hypothesis we made on the possibility of estimating brix° from appearance.

5.2.5 Multi-task

Starting from the insights of the experiments described in the previous Section, we evaluate the multi-task architecture on some of the splits described in Table 4, i.e. Season21ToSeason22Phone, Season21ToSeason22Phone_WB, MixedSeason and MixedSeason_WB. This choice was the consequence of needing more data to properly train a deep neural network and to test covariate shift using different devices. Again, we tried to find an answer about the effects of WB images on performance.

Each of these splits defines a training and testing sub-division, so the results are not directly comparable between Season21ToSeason22Phone and MixedSeason splits, if not only by generic considerations. Two sets of augmentations were used during training: the basic one had only horizontal flips and Gaussian noise; the alternative one also added random perspective and random crops. These two choices have been evaluated as a binary hyper-parameter during the parameter sweep. As described in Section 3.4.3, the multi-task loss has two hyperparameters (called alpha and beta). In addition to them, the standard hyper-parameters have been evaluated (learning rate, momentum, batch size), together with more specific ones, such as image dimension, and the use of a pre-trained backbone (resnet18).

Since a multi-task problem such as this has different, and possibly contrasting objectives, to perform a sweep a single metric had to be designed. We considered the context of the problem to design one. The validation loss is not very informative on the general performance of the two heads, but a metric including the colour loss and the brix loss could work. However, we considered penalising the runs that were above the error limits given in the specification (accuracy above a certain threshold and brix error less than a certain MAE threshold), or less. The penalization scheme is based on a hinge loss as described in the following equations describing the validation score s:

$$
\begin{aligned}
s &= \eta(1 - A_c) + \lambda L_{brix} + L_{CH} + L_{BH} \\
L_{CH} &= \max(0, -\eta(A_c - t_c))^{\rho} \\
L_{BH} &= \max(0, L_{brix} - t_{brix}))^{\rho}
\end{aligned}
$$

where t_brix and t_c are the thresholds (in our experiments 2.5 and 0.8, respectively), eta is a normalising factor to balance the contribution of accuracy vs MAE error, and it is simply t_brix/t_c,

and rho is the degree of the hinge losses, in our case always 2. Even if not a perfect measure, as the following discussion will show, this score was enough to steer the hyperparameter optimization to reasonable and interpretable results.

The results on parameter relevance of the total 584 runs are summarised in Fig. 25 and 26, While the results for the best run in the sweep are shown in Fig. 27, 28 and 29. In particular, from these three images, we can notice two important aspects. First, the best accuracy and the best brix error do not correspond to the same epoch. This means that, while training together the two problems gives some advantage (still to be studied with an ablation study), at the same time the nature of the loss leads to a sort of alternating optimization, i.e. when the loss on the brix degree is low, then the backpropagation concentrates on lowering the colour loss, or vice-versa, but it is no easy to find a common good solution for both. This does not mean that the solution does not exist, only that this training procedure does not easily converge to a common solution and feature representation.

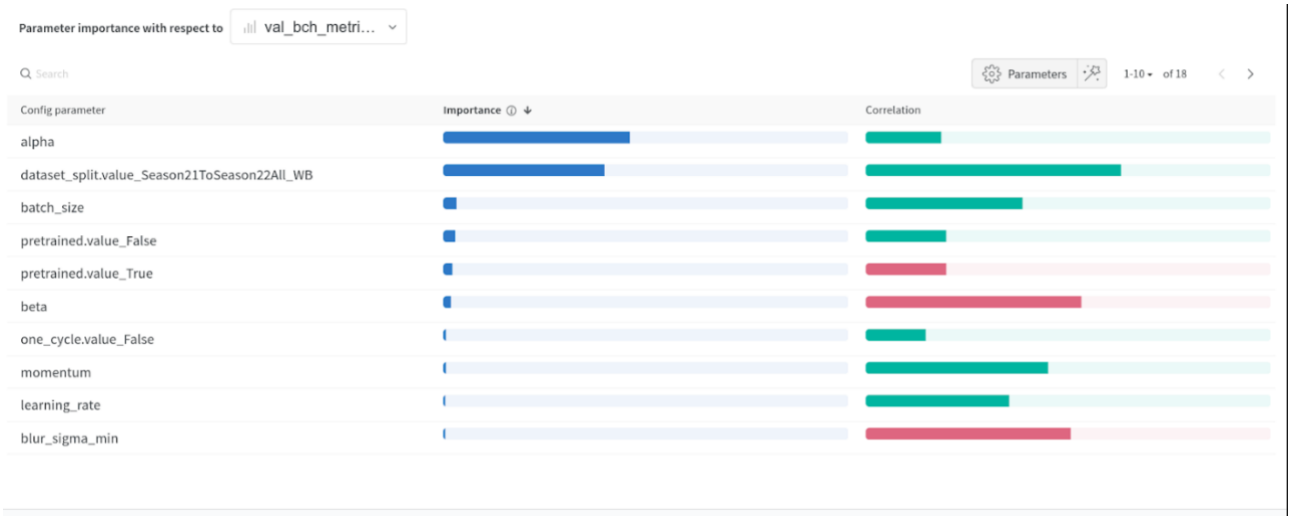

Figure 25 Parameter importance plot for SeasontoSeasnonPhone experiments. From this diagram we see that both alpha and beta should be high, and that definitely the images should be white balanced. Pretraining seems to be important, even if on imagenet data.

The second observation is that, if we neglect the computational constraints and want to maximise the performance in both brix and colour estimation, we have only to save the snapshot or the network at two different epochs, according to their respective minima. This is important because the brix MAE and accuracy for the best run are very interesting results (1.08 MAE and 77%).

This project is funded by the European Union G.A No 101016906

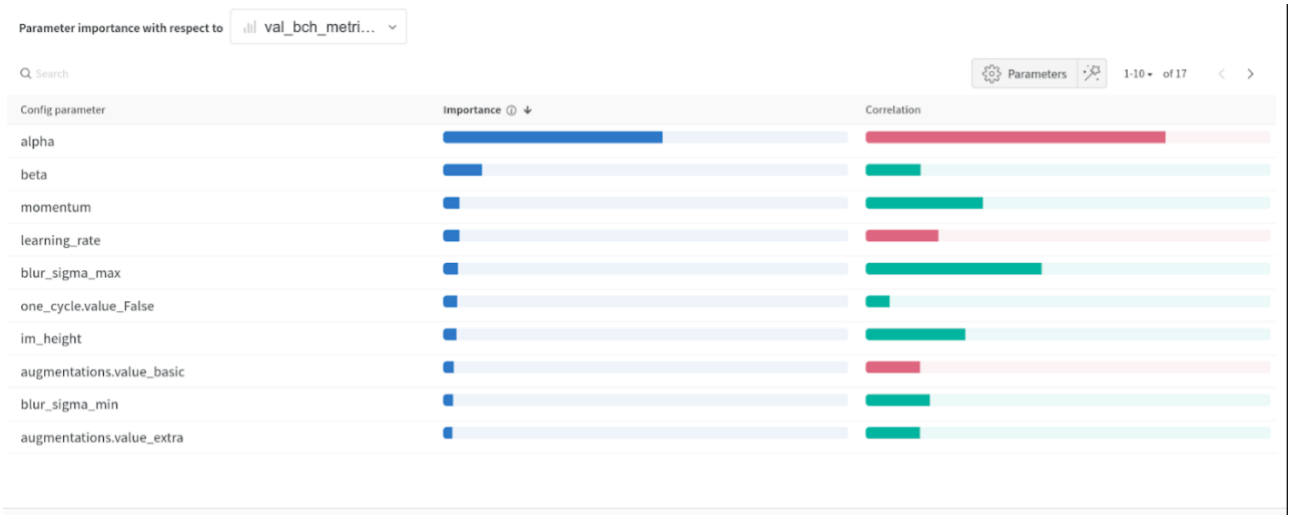

Figure 26 Accuracy score for the best run of the multi-task architecture. The highest value of 0.77 happens at epoch 45.

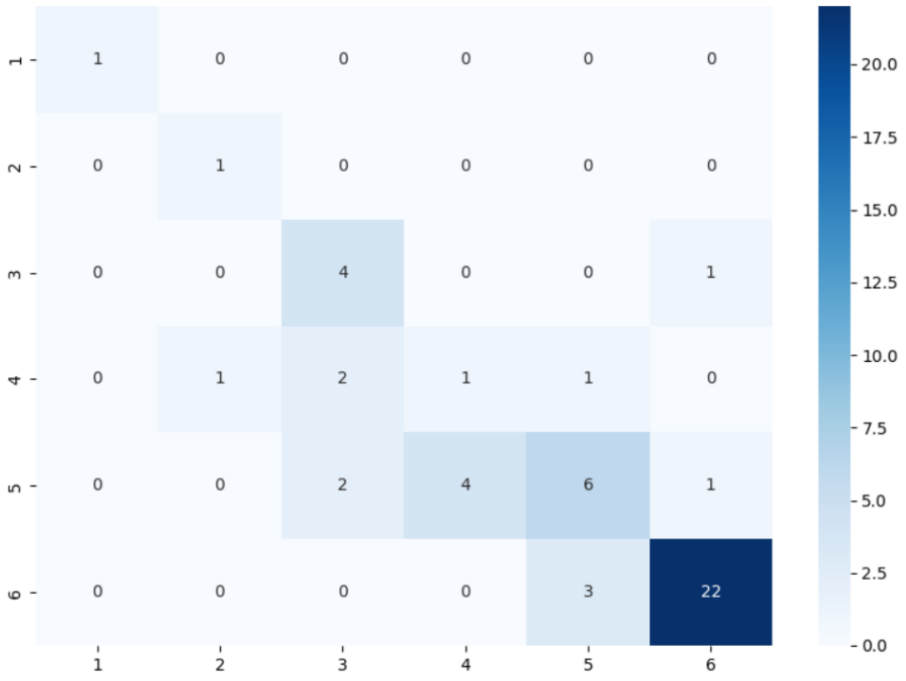

Figure 27 Confusion matrix on the validation set of MixedSeason for the best run in all the sweep. Errors that cause a wrong harvesting decision are 7/44, around 15%, achieving the performance target fixed for this parameter.

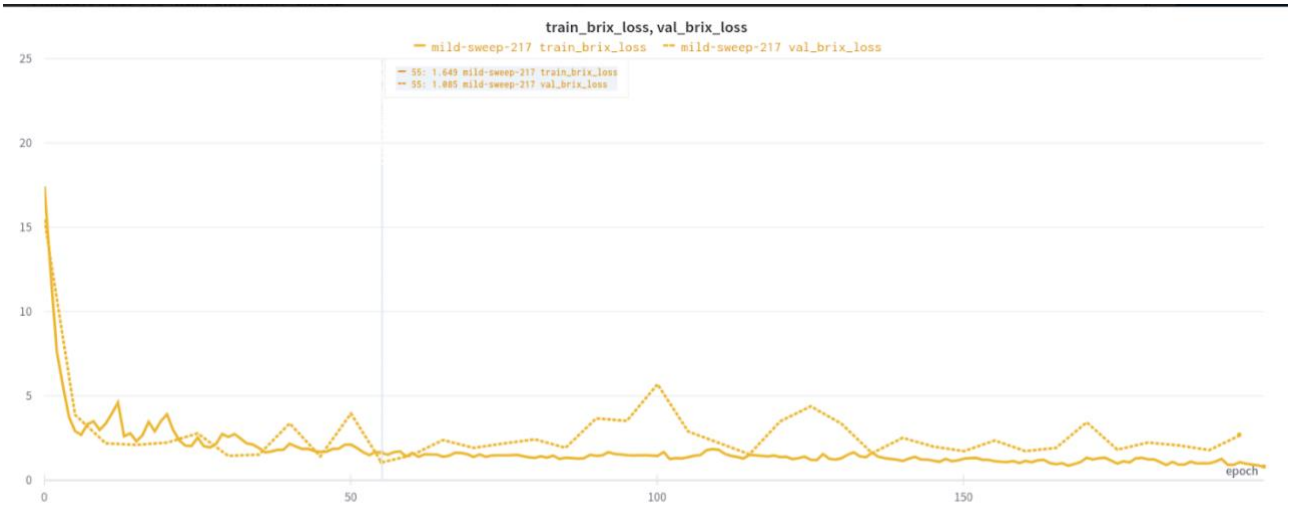

Figure 28 Accuracy score for the best run of the multi-task architecture. The highest value of 0.77 happens at epoch 45.

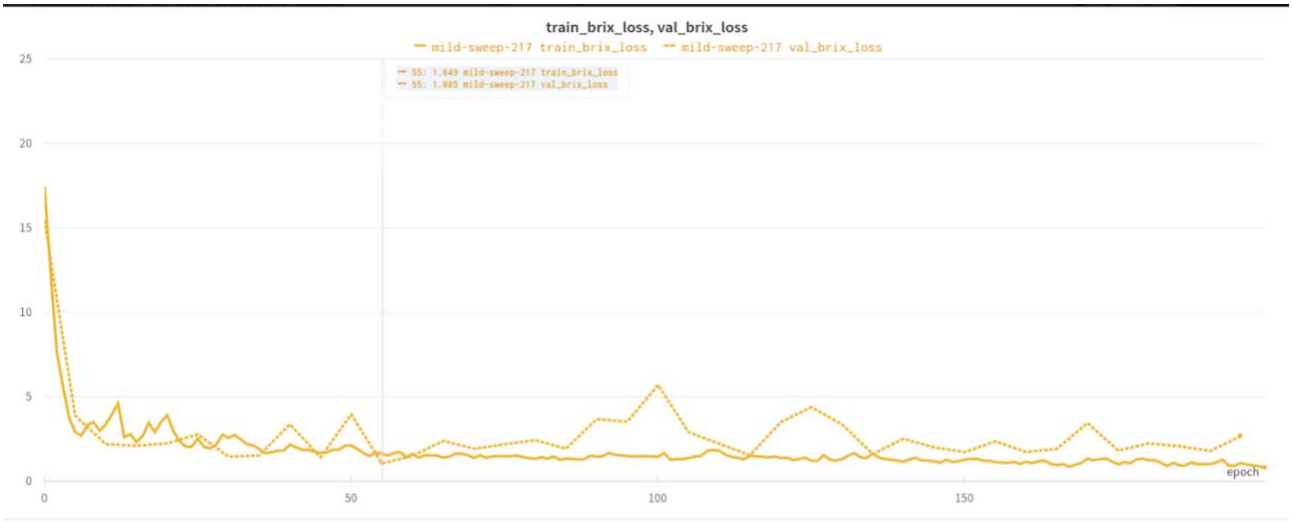

Figure 29 Brix° MAE for the best run of the multi-task architecture. The lowest value of 1.08 happens at epoch 55.

5.3 Size and shape estimation validation

To validate the method proposed, we first tried measuring rigid objects of known shape, like spheres and boxes, by manually sampling the distances to measure.

Wooden Box:

- GT length: 79 mm -> Predicted length: 79 mm
- GT diagonal: 87 mm -> Predicted diagonal: 87 mm

For those kinds of objects, in static conditions with controlled illumination, the results were millimetre perfect, and therefore we moved on to plastic grapes, using the detector.

Plastic grapes:

1. Left grape
    - GT height: 148 mm -> Predicted height: 149 mm
    - GT width: 60 mm -> Predicted width: 57 mm

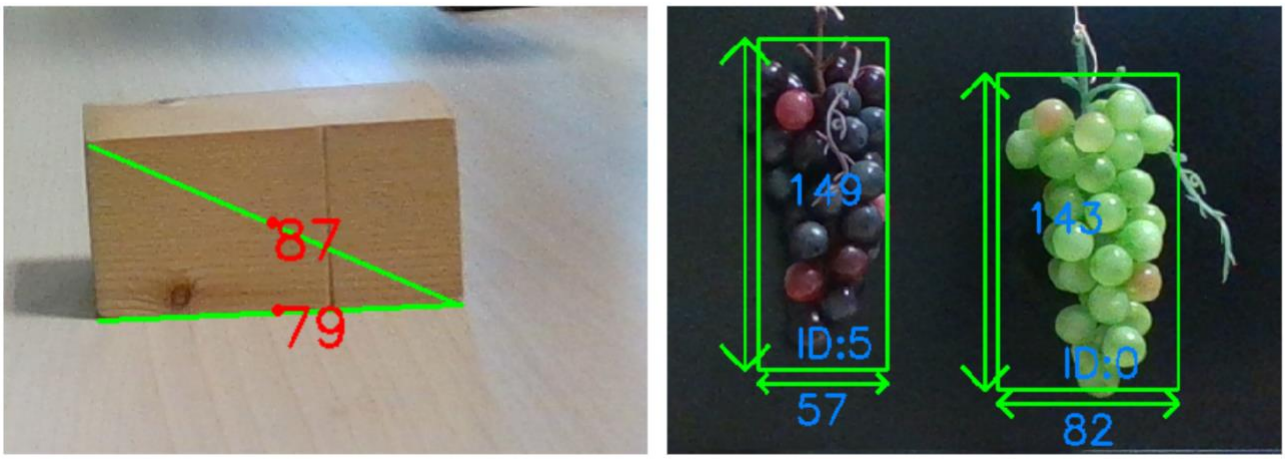

Figure 30 ***Setup for verification of depth and size estimation accuracy at bunch level.***

2. Right grape
    - GT height: 145 mm -> Predicted height: 143 mm
    - GT width: 80 mm -> Predicted width: 82 mm

Also, in this case, the results were promising but not perfect due to the detector inaccuracies, leading to a measurement error lower than 5%, thoroughly validating the methodology proposed.

We also tested the berry size estimation module on plastic grapes. Since the critical value is not the size of single berries but the average size of all the berries on the grape bunch. For this reason, to obtain the ground truth value, we sampled the size of single berries on the test grape to then keep the average. Apart from the mean value, another significant value is the standard deviation since it provides the distribution of the size of the berries in the grapes.

Plastic grape:

- GT berry diameter: 20 mm -> Predicted diameter: 19.26 mm
- Standard deviation: 3.30 mm

This project is funded by the European Union G.A No 101016906

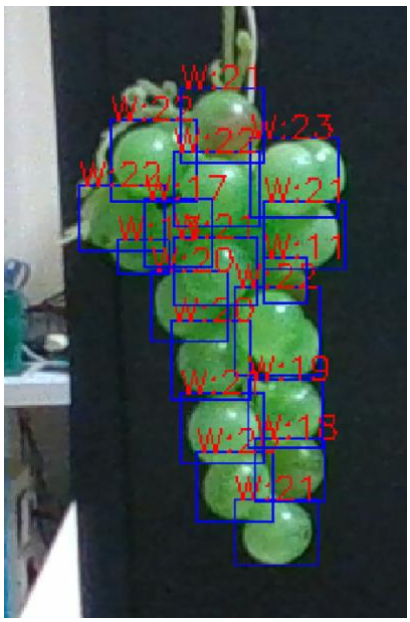

Figure 31 Example of berry size estimation.

In percentage points, this is a 16% standard deviation. Clearly, berry size in this case is computed on round berries. Computing size estimation on berries with different berry shapes will add some error, mainly due to the perspective and orientations of the berries. Modelling the 3D shape of a non-round berry is a very complex issue compared to the purpose of the project, where a 30% MAE error is tolerated. Nonetheless, during T9.3, field validation for agronomic-oriented perception, some modifications will be done to reduce the effect of non-round shape if the error will raise beyond the required limits.

This project is funded by the European Union G.A No 101016906

6 Conclusion

The report gives technical information about the problem of quality estimation of table grapes using RGB-D inputs. Data collection campaigns during the two summer seasons of 2021 and 2022 have been described, together with the details of the post-processing and labelling of training and validation data.

Methods for the estimation of all four quality parameters have been presented, and all of them have been validated on datasets collected in the experimental field. The results achieved are the following:

- Anomaly detection for harvesting (binary decision): 87% precision
- Colour for harvesting (binary decision): 84% accuracy
- Brix° estimation: 1.08 MAE
- Berry size MAE: 3.7% average error with 16%

In all cases the performance target KPIs are outperformed.

From the point of view of computational costs of the algorithms, we did not run formal tests, since these depend also on the final hw that will be mounted on the manipulator. As a reference for the reader, if every module of the QEM is running at the same time, on a Nvidia Jetson Orin the system goes at 1Hz. However, no effort was invested in optimization so far. All these questions will be addressed as part of the work to be done for WP9.

7 Bibliography

[1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, April 1997, doi: 10.1109/4235.585893.

[2] Innotescus software: main website no more available. General description at https://sourceforge.net/software/product/Innotescus/

[3] CVAT Annotation Tool: https://github.com/opencv/cvat

[4] Robot Operating System: https://www.ros.org/

[5] Ciarfuglia, Thomas A., et al. "Weakly and semi-supervised detection, segmentation and tracking of table grapes with limited and noisy data." Computers and Electronics in Agriculture 205 (2023): 107624.

[6] Rawpy package: https://pypi.org/project/rawpy/

[7] YOLOv5: https://github.com/ultralytics/yolov5

[8] Zabawa, Laura, et al. "Detection of single grapevine berries in images using fully convolutional neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2019.

[9] Weight & Biases: https://wandb.ai/site

[10] Data Version Control (DVC): https://dvc.org

[11] Jira: https://www.atlassian.com/software/jira

[12] YouTrack: https://www.jetbrains.com/youtrack