applied sciences

MDPI

Article

A Cobot in the Vineyard: Computer Vision for Smart Chemicals Spraying

Claudio Tomazzoli [1,†], Andrea Ponza [2,†], Matteo Cristani [1,†], Francesco Olivieri [3,†] and Simone Scannapieco [2,*,†]

[1] Department of Computer Science, University of Verona, 37134 Verona, Italy; claudio.tomazzoli@univr.it (C.T.); matteo.cristani@univr.it (M.C.)
[2] Real T S.R.L., 37131 Verona, Italy; andrea.ponza@realt.it
[3] Independent Researcher, Brisbane, QLD 4121, Australia; francesco.olivieriphd@gmail.com
[*] Correspondence: simone.scannapieco@realt.it; Tel.: +39-3486539501
[†] These authors contributed equally to this work.

Abstract: Precision agriculture (PA) is a management concept that makes use of digital techniques to monitor and optimise agricultural production processes and represents a field of growing economic and social importance. Within this area of knowledge, there is a topic not yet fully explored: outlining a road map towards the definition of an affordable cobot solution (i.e., a low-cost robot able to safely coexist with humans) able to perform automatic chemical treatments. The present study narrows its scope to viticulture technologies, and targets small/medium-sized winemakers and producers, for whom innovative technological advancements in the production chain are often precluded by financial factors. The aim is to detail the realization of such an integrated solution and to discuss the promising results achieved. The results of this study are: (i) The definition of a methodology for integrating a cobot in the process of grape chemicals spraying under the constraints of a *low-cost* apparatus; (ii) the realization of a *proof-of-concept* of such a cobotic system; (iii) the *experimental analysis* of the visual apparatus of this system in an indoor and outdoor controlled environment as well as in the field.

Keywords: collaborative robotics; precision agriculture; cyber–physical systems; artificial intelligence; machine learning; deep learning; computer vision

check for updates

Citation: Tomazzoli, C.; Ponza, A.; Cristani, M.; Olivieri, F.; Scannapieco, S. A Cobot in the Vineyard: Computer Vision for Smart Chemicals Spraying. *Appl. Sci.* **2024**, *14*, 3777. https://doi.org/10.3390/app14093777

Academic Editor: Andrea Prati

Received: 28 February 2024
Revised: 22 April 2024
Accepted: 26 April 2024
Published: 28 April 2024

1. Introduction

With the development of the global economy, novel agricultural approaches to prevent growing problems and risks of disease transmission have gained momentum. Emerging epidemics and pests in plants as well as infectious diseases in terrestrial animals have a substantial economic and logistic impact on agricultural and forest productivity, trade and public health. Therefore, the need for on-field treatments, which should be concentrated both in time and space to avoid waste of money and pollution whilst maintaining farm productivity, is great.

Precision agriculture (PA) is an emerging trend that promises an increase in the quantity and quality of agricultural output while using less input—like water, energy, fertilisers and pesticides; the aim is to produce more and better food while reducing costs and limiting the environmental impact. One key enabling technology for PA, among all those inherited from Industry 4.0 [1], is that of the cyber–physical system (CPS), intended as an ICT-based system purposefully built to interact with other CPSs and humans.

The focus of the present investigation is on the development of a CPS able to assist winemakers—in particular, those present in the north-eastern regions of Italy as the first experimental targets—which are typically small–medium enterprises (SMEs) with specific constraints. Since vineyards in this context are relatively small-sized when compared to other farming activities, and most SMEs are not financially strong enough to afford full-robotic systems for economic and logistic reasons, a *low-cost technology integration* should be pursued.

The current research effort in a viticultural context should be envisioned into a broader long-term objective; that is to say, the creation of an affordable agricultural collaborative robot (cobot) able to carry out basic tasks in a rural environment. The agricultural cobot would intervene by helping on-field operators in their more repeatable tasks, and surely the most physically endangering and demanding ones. The first case study, which is proper of a vineyard setting, involved chemical spraying on grape clusters (e.g., fungicides), and precision treatments where grape buds, inflorescence and clusters should be avoided (e.g., weed herbicides). The nature of these and other similar type of actions inevitably requires *vision capabilities* to be integrated into the CPS.

The agenda for the present dissertation consists of the following three research goals:

- The definition of a methodology for integrating a cobot in the process of grape chemicals spraying under the constraints of a *low-cost* apparatus;
- The realization of a *proof-of-concept* of such a cobotic system;
- The *experimental analysis* of the visual apparatus of this system in an indoor and outdoor controlled environment as well as in the field.

The design of a cobot usage protocol is described, and the experimental apparatus for the whole technological solution is presented. The adopted computer vision (CV) system shall be widely described and featured with experimental results, while the cobot motion planning and control apparatus is only sketched and shall be tested experimentally in further work.

The paper is organized as follows: Section 2 highlights the state of the art and related work about robotic solutions and CV techniques for agriculture technology. Section 3 examines the enabling technologies and macro-components necessary for the proposed solution. Section 4 shows the experimental results achieved. In particular, Section 4.1 focuses on the implementation of the vision model. Section 4.2 is dedicated to explaining the simulation phases restricted to pure vision capabilities, and critically analyzes the performance metrics against other recent solutions. Section 4.3 is a brief report of the first cobotic tests conducted both in controlled environments and in the vineyard. Finally, Section 5 outlines several open issues and research paths left for future work.

2. Background

Agricultural robot technologies are nowadays one of the main catalysts for smart industry [2], but in most cases, the research and application projects focus on the development of individual specific-purpose machines [3] or delve deeply into the topic of advanced control and obstacle avoidance. Moreover, when required, recognition capabilities based on CV are often considered outside actual robotic integration.

Among the very few solutions developed in recent years in viticulture contexts, one noteworthy example is introduced in [4] as a full-robotic monitoring system based on images taken from a camera mounted on a mobile robot, which localizes itself in the field using simultaneous localization and mapping (SLAM). Nevertheless, neither a full adoption nor a partial integration of such solutions for the present case study is feasible, since none of them have been framed in a perspective of *low-cost* architecture development. In fact, although this aspect of applications of cobotic solutions in agriculture is of crucial importance for the diffusion of the mentioned means, especially in those contexts where it is essential from a social viewpoint [5], little consideration has been spent in the subject matter. In particular, in [6], the authors deal with aspects related to general architecture costs, while in [7], the authors discuss low-cost camera applications.

For a CV-assisted cobotic solution to comply to economic constraints, three crucial problems must be addressed: (a) machine vision for fruit (in particular, grape clusters) recognition, (b) chemicals spraying and (c) coordination among humans and robots. In what follows, a brief outline of the state of the art in the literature is reported for each devised problem.

2.1. Machine Vision Capabilities

Machine vision applications in agriculture has been one of the most studied topics and, generally speaking, one of the most successful and promising fields in the smart agriculture domain [8–10]. Among the pure CV methods in agricultural contexts, the most relevant case for our investigation is the application to viticulture (devised in Section 4). On the other hand, noticeable exploratory analyses have been conducted on wheat farming [11]— to provide real-time spatial information of the crop and weeds, where conventional methods work undesirably in terms of segmentation accuracy or execution speed—and on the ryegrass location [12]—where a geometric approach has been shown to be effective in the determination of inter-row ryegrass weeds in a wheat field.

Many studies in the field consider more refined types of vision challenges other than simple detection or spatial distribution tasks. Particularly relevant to the present research is the automatic determination of the maturation levels of the same fruits, which has been faced by using image analysis techniques as a general problem [13] and with respect to specific fruits such as the persimmon [14].

2.2. Spraying Capabilities

Efficient irrigation has been investigated on large scale yards [15] as an optimisation problem, where the goal is to find the best path to traverse a graph where each vertex has an associated reward and each edge has a defined cost (i.e., a typical approach of reinforcement learning). Analogous tasks have also been explored in small scale, i.e., gardening [16], where a framework to decide the correct amount of water and chemicals for each plant is implemented.

Vineyard sprayer robots have also been investigated, in particular, in [17] and other studies. The platform diffuses chemicals in both directions, orthogonal to the robot's trajectory, while moving in the yard. In the same work, an interesting navigation technique has been devised using a localization algorithm based on the maximum sum of probabilities intersections for the optimal fusion of inexpensive navigation sensor data while maintaining a moderate budget. While dealing with the problem of chemicals treatment distribution, the topic of precision spot-spraying for robotic applications has also been dealt with by [18] describing both the airflow structure and the spray coverage.

2.3. Cobotic Capabilities

Interaction with human operators in the fields, specifically human–robot interaction user interface design and usability evaluation, has been explored only in part so far. Some of the limitations identified pertain to the lack of peripheral vision, the amount of operator time required to perform the movement (pan-tilt, zoom-in and zoom-out) needed to identify targets and the calibration of the peripheral screen devices due to sunlight [19].

3. Materials and Methods

3.1. Enabling Technologies

An initial assessment of the state of the art provided potential needs for a CPS in the context of PA, regarding the following three macro-components:

- Manipulation (Section 3.1.1);
- Recognition (Section 3.1.2);
- Movement.

These components shall be managed via an orchestrator, which allows for inter-communication and a managed lifecycle (Section 3.1.3).

3.1.1. Manipulation

Manipulation describes the possibility for a CPS to interact with the surrounding environment by picking and placing objects in its environment within its workspace.

Manipulation is a hot topic since the design of the first robotic apparatus in the 1940s [20,21]. Over the years, a general consensus emerged about the basic structural shape of a manipulator, i.e., a *body and arm assembly* guaranteeing spatial positioning, a *wrist assembly* conferring dexterity and an *end effector* capable of orientating itself with some degree of freedom.

Coordination of every robot joint's movement is fundamental to manipulate an object towards a precise configuration. Commands are provided in the joint space or in the Cartesian space; that is to say, they can be given directly to the motors moving the single robot joints with angular coordinates or with spatial coordinates to reach a predefined position of the end effector (potentially with multiple orientations).

Inverse kinematics allows the calculation of joint space coordinates to obtain a position in the workspace.

Explaining in detail how different body and arm assemblies and wrist assemblies configurations are combined to influence manipulation is not among the goals of the present work. Briefly speaking, these different combinations modify the cardinality of solutions for the inverse kinematic problem and, thus, the possible configurations to reach the target pose. Upon reaching the manipulation pose, the end effector can be actuated for the desired action to be executed.

Manipulation allows direct interaction with the CPS' environment and with humans. The dexterity of how robots move in space allow them to reach goal configurations where the end effector is used; this is why manipulation is crucial to the realization of a CPS.

3.1.2. Recognition

Recognition describes the possibility for a CPS to see obstacles while moving in space, and to recognize its surroundings and what it could manipulate at the same time. All of these actions require some *vision* capabilities to be provided to an otherwise blind system. Vision is only one way to give senses to robots—together with touch, hearing and equilibrium (particularly regarding the change between stasis and acceleration)—and make them more similar to humans. CV is the interdisciplinary field dealing with supplying high-level comprehension to a computer via digital images and videos. Born in the 1960s [22], CV finds a key development with the introduction of deep learning (DL) and neural networks.

Sight can be obtained via traditional cameras, sonars, lasers or RFID. However, the more up-to-date strategy for a chance at recognizing interactive objects is using a classifier based on convolutional neural networks (CNNs). The most desirable way to implement this technology is using cameras (to give the neural network a field-of-view to evaluate the presence and position of objects—in this case study, grape clusters), whereas obstacle recognition and avoidance can also be solely based on sonars and lasers.

Recognition represents a fundamental requirement to introduce a cobot in an agricultural company, as it allows the CPS to acquire the following two separate but complementary functions:

Interactive objects recognition: being able to discern the class of an object and, therefore, to determine whether the CPS can interact with it via manipulation (the object can be picked, actions can be performed with or on it) or with movement (the object is a charging base, the object represents the goal coordinates of the mobile base and so on);

Obstacle recognition: to execute manipulation or movement trajectories, the CPS understands that certain portions of space must be avoided not to incur into collisions.

3.1.3. Orchestration

A CPS must behave in predefined cycles and react to unpredicted events in a robust and effective way. To actually realize such a behaviour, a unit is needed that manages various physical and logical components, in order to achieve a preset goal. The orchestrator grants the ability to coordinate the other capabilities of a CPS and make them intercommunicate.

It basically is middleware for robotics, i.e., an agent supporting a distributed architecture that (i) integrates processes and services residing on multiple technologies and architectures, and (ii) works between the operating system and the user's application, as reported in Figure 1.

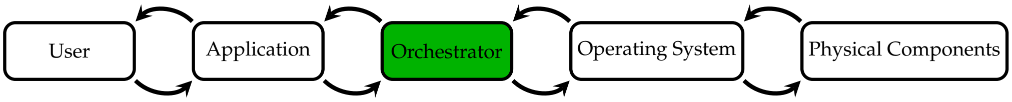


Figure 1. The orchestrator allows hardware and software intercommunication.

Over the time, many robotics middleware projects have been developed, such as the following:

ROS: it is an open source meta-operating system providing services like hardware abstraction, low-level device control, commonplace functionality implementation, message passing between processes and package management, together with the instruments and libraries required to obtain, write, compile and run code on different computers;

MRPT: it is an open source, multiplatform, C++ library aiming to help robotic researchers with designing and implementing algorithms like SLAM, CV and path planning with obstacles avoidance;

YARP: it supports building control systems for robots as collections of programs that communicate in a peer-to-peer architecture, with extensible and interchangeable types of connections (tcp, udp, multicast, local, mjpg-over-http, tcpros, . . .) and the strategic goal of increasing the life span of robotic software projects.

Being at times preferable to develop and integrate systems independently (especially in pilot projects), not all robotic systems use middleware. This, however, becomes an overbearing task when attempting to create robust, maintainable projects that manage complexity in an intelligent way.

3.2. Designed Solution

The analysis of the enabling technologies reported in Section 3.1 led to the following cobotic experimental solution:

- An affordable anthropomorphic manipulator with as many degrees of freedom as possible;
- A highly responsive CNN classifier to recognize grape clusters, obstacles and their positions from visual input acquired by adequate sensors;
- ROS orchestration;
- A low-cost cobotic processor that integrates vision-related tasks of the CPS into orchestration.

3.2.1. Hardware Specifications

The collaborative robot model Panda (from Franka Emika GmbH) was chosen as development manipulator, mainly for its straightforward integration with ROS. The high abstraction level of ROS to accommodate to different hardware solutions does not preclude the possibility to seamlessly shift to different cobotic choices in subsequent steps.

The end effector of choice for the current proof of technology is the Franka Hand (Figure 2) provided with the Panda arm, while subsequent implementations could use dedicated end actuators available in the market (like the one in Figure 3 for spraying). For the first round of tests, a generic perfume sampler was used in place of an actual nutrients dispenser.

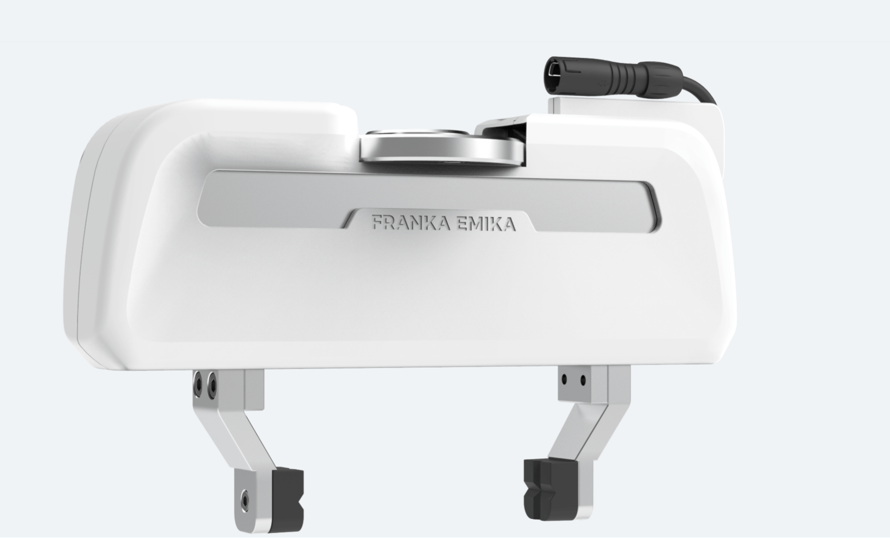


Figure 2. Franka Emika Panda's end effector. Source: https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka_Emika_Hand_01.jpg, accessed on 25 April 2024.

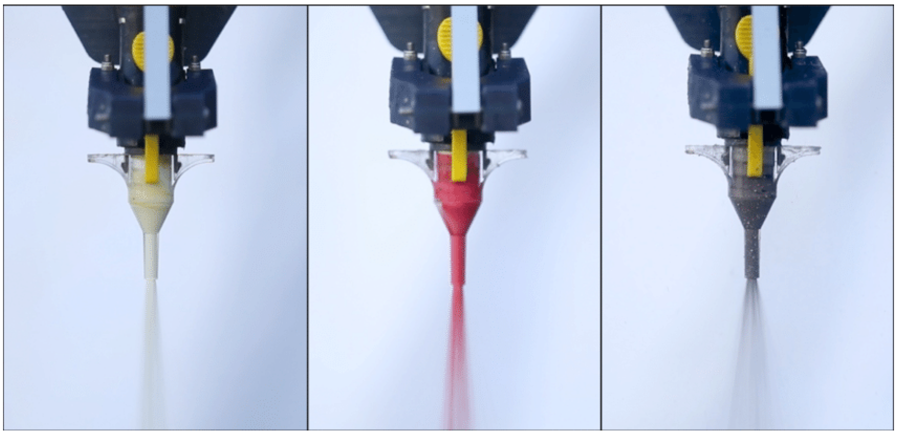


Figure 3. Example of a sprayer end effector. Source: https://www.researchgate.net/publication/313497501/figure/fig18/AS:614364813983767@1523487394998/Variable-color-output-from-the-foam-spray-end-effector-is-seen.png, accessed on 25 April 2024.

An Intel® RealSense™ D435 Depth Camera (https://www.intelrealsense.com/depth-camera-d435/, acquired in Verona, Italy on https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435.html, accessed on 25 April 2024) was chosen in a unifying effort to accommodate collaborative robotics with CV (in particular, robotic planning assisted by visual recognition). This enables scene analysis, obstacle recognition by working in three dimensions and the usage of *neural networks* as visual classifiers.

In order to provide a low-cost architecture, one of the main efforts was to consider the integration of mid-range, general-purpose cobot processors rather than high-end, edge computing processors specifically engineered for AI-assisted robotic tasks (as, for instance, the family NVIDIA Jetson Xavier/Orin—https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, accessed on 25 April 2024). On the other hand, the cobot processor must comply to crucial constraints regarding data communication with the arm controller. The technical specifications of the processor adopted are reported in Table 1.

Given the obvious imbalance between realization costs and expected performances, a systematic work in software optimization has been conducted, especially in the definition of the neural recognition layer (see Section 3.2.2).

Table 1. Specifications of the cobot processor. NUC Kit, CPU, NIC and LAN acquired in Verona, Italy on https://www.amazon.it/INTEL-Barebone-BLKNUC7i7DNK2E-Core-i7-8650U/dp/B07BZCY271. RAM acquired in Verona, Italy on https://it.crucial.com/memory/ddr4/ct8g4sfs824a. OS publicly available on https://releases.ubuntu.com/16.04/.

Model	Intel® NUC Kit NUC7i7DNKE
OS	Ubuntu 16.04 64 bit, Open Source
CPU	Intel® i7-8650U (8 M Cache, up to 4.20 GHz)
RAM	Crucial Single Rank 16 GB DDR4, 2400 MT/s
NIC	Intel® Dual-Band Wireless-AC 8265 Wi-Fi
LAN	Intel® i219-LM, up to 1000 BASE-T
GPU	—

3.2.2. Software Specifications

Software sub-components and libraries are required that permit the adoption of neural networks in the control cycle (from sensors to motion controllers and back to actuators), in order to properly recognize the arm's surroundings. In this regard, the neural network training quota has prevailed over other components in terms of development time.

OpenCL Caffe has been chosen as the codification standard for the neural network (for both architectural representation and weight codification). As a Caffe fork, OpenCL Caffe is a DL framework adapted to work with or without a GPU, thanks to the OpenCL libraries, and particularly optimized for GPU-less, Intel-based architectures. The experimental line of neural network training actually involved an additional framework named Darknet, whose characteristics are specified in Section 4.1. The neural network development in the Darknet standard, with subsequent OpenCL optimization and translation in the OpenCL Caffe standard, resulted in an optimal pipeline to obtain a component that can be easily integrated with other software components.

4. Experimental Results and Discussion

4.1. Vision Capabilities Implementation

The ideation, training, validation and test cycle for neural networks has been managed with the help of Darknet, an open source framework for the creation of CNNs in C and CUDA. Darknet has become a standard thanks to the engineering of the YOLO (You Only Look Once) family of systems, state-of-the-art lightweight neural networks architectures for real-time object detection. Starting from YOLOv1, the framework has been enriched with ever more sophisticated variants over the course of multiple years (YOLOv2, YOLOv3, up to YOLOv4 and Scaled-YOLOv4). The fourth version of YOLO was the last to be written in Darknet in 2020; the AI company Ultralytics decided to further refine the YOLO architecture (up to YOLOv8 in 2023), but favored the PyTorch technology over Darknet.

In the first experimental phase, two variants of model YOLOv2 have been adopted, i.e., YOLOv2-416 and YOLOv2-608. Although quite outdated, the choice is justified by the following:

- Given the high dispersion rate of most chemicals in agricultural environments, the action of "precision spraying" does not necessarily require top-notch performances (for instance, in regard to IoU) to be considered effective. In our preliminary case study, YOLOv2 showed adequate performances with respect to the recognition of grape clusters as well as bounding box precision, with no need to scale to more refined versions (see Sections 4.1.2 and 4.2.4). Nonetheless, a comparative analysis with the YOLOv3/v4 performances is currently ongoing, and their adoption is not excluded in the foreseeable future;
- YOLOv2 allows a close-to-real-time performance for inference, even with GPU-less compute units such as the one adopted in our case;
- The (re-)training of YOLOv2 is significantly faster when compared to a more recent version of the architecture due to the relative simplicity of the underlying backbone;

- Being an open source framework, Darknet implementations of YOLO (i.e., up to YOLOv4) are license-free by construction, whilst embedding Ultralytics software and AI models in commercial products and applications strongly recommends an enterprise licensing, unless the open source requirements of AGPL-3.0 are met. This aspect must be considered in view of a trade-off between cost reduction and performance.

4.1.1. Dataset

Rather than creating an image dataset from scratch by acquiring high-resolution images (as, for instance, in [23,24]) or stereo images [25] directly on the field, the Embrapa Wine Grape Instance Segmentation Dataset (WGISD, https://github.com/thsant/wgisd, accessed on 22 April 2024) has been considered as the starting point for our investigation. This is an approach that guarantees reproducibility and, simultaneously, results that perform well in terms of effectiveness, especially in the field. The dataset has been augmented accordingly with this purpose, as shown below.

The WGISD was purposefully created for use in tasks concerning image-based monitoring and robotics in a vinicultural setting (see, among others, [26]), and it provides visual instances of five different grape varieties. The dataset shows a good variance with respect to cluster pose, illumination, lens focus and it includes genetic and phenologic variations that impact the shape, the color and the compactness of the bunches.

The original dataset contains 300 images with 4432 grape clusters identified by the same number of bounding boxes (object detection) and 2020 bunches identified by binary masks (semantic segmentation). The dataset subdivision between training, validation and testing for WGISD (for images and bounding boxes) is given in the first row of Table 2.

Table 2. Dataset partition for grapes recognition.

	Images		**Bounding Box**	
	Train/Val	**Test**	**Train/Val**	**Test**
WGISD	242	58	3581	850
WA-WGISD	1694	58	25,067	850

Dataset Aggregation and Augmentation

Following the current state of the art in the literature, the WGISD dataset has been aggregated with an additional general purpose dataset for the following reasons:

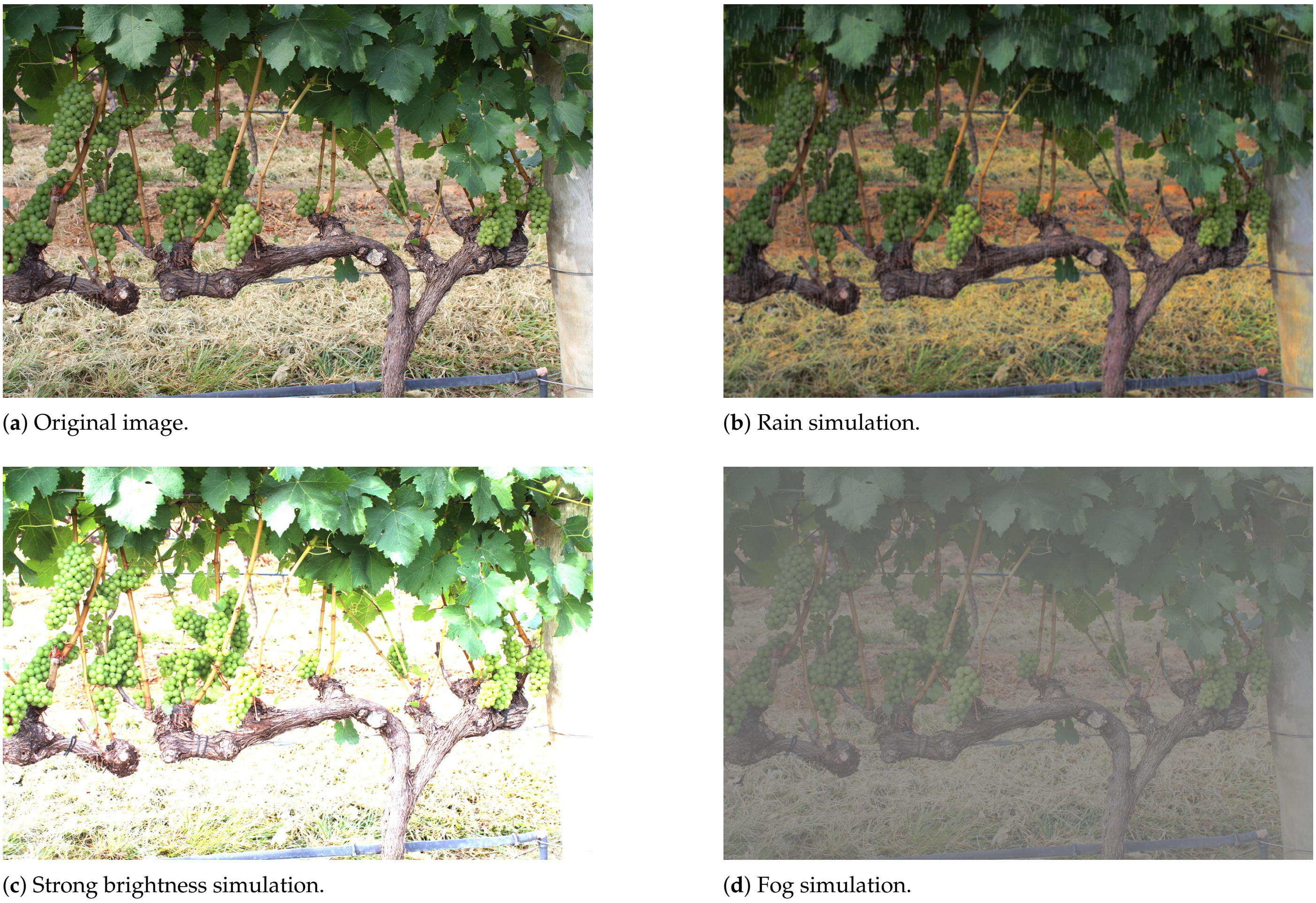
- Radically avoiding overfitting problems, usually determined by the construction of neural networks for the recognition of a single class;
- To be able to follow an analysis strategy of the average precision percentage on the "grapes" class based on the average precision of other classes (in essence, check how well the trained neural network behaves with the new "grapes" class compared to all other recognizable classes);
- To exploit inference results of grape-unrelated classes and further empower the mechanism of obstacle avoidance (e.g., applying different security protocols based on the *nature* of an obstacle—a person, another machine in the field and so on).

In order not to create a neural network with overabundant knowledge that could excessively impact each training session, the choice fell to the PASCAL VOC 2007+2012 dataset (http://host.robots.ox.ac.uk/pascal/VOC/, accessed on 22 April 2024), which provides a catalog of 20 classes, varying from vehicles, to home decor items and animals. Table 3 shows the subdivision of training/validation and testing regarding images and bounding boxes in the PASCAL VOC 2007+2012 dataset.

Given the clear mismatch between the number of images to train/validate on WGISD and the single class average on PASCAL VOC 2007+2012, the original WGISD dataset underwent an image augmentation process. Several strategies may be adopted in this regard, for instance, which simulate camera focus, noise or dirty lenses on randomly selected images in the source dataset [26], or by simply integrating additional images from other datasets to obtain a balance for the specific class (like GrapeCS-ML added to the Open Image Dataset v6 restricted to the "grapes" class [27]). The innovative augmentation technique proposed hereafter enriches the dataset by simulating *different weather conditions* in the vineyard (for example, intense sun, fog, rain, haze and so on). Examples of image augmentation for the WGISD dataset with atmospheric simulations are given in Figure 4. In this way, the resulting vision model is expected to be more robust even in strongly over- or under-exposed scenarios, where the lack of detail and luminance/chrominance distortions considerably undermine the inference process. This technique bears a strong resemblance and shares the same objectives of several cutting-edge studies in image enhancement and restoration methods in CV (among others, [28–30] for intelligent image de-hazing and [31,32] for image restoration under extreme low-light conditions).

Table 3. Dataset subdivision of PASCAL VOC 2007+2012.

	Images		Bounding Box	
	Train/Val	Test	Train/Val	Test
PASCAL VOC 2007	5011	4952	12,608	12,032
PASCAL VOC 2012	11,540	11,540	27,450	27,450
PASCAL VOC 2007+2012	16,551	16,492	40,058	39,482
Single class average	827.55	824.6	2002.9	1974.1


(**a**) Original image.

(**b**) Rain simulation.

(**c**) Strong brightness simulation.

(**d**) Fog simulation.

Figure 4. Examples of image augmentation for the WGISD dataset.

The WGISD dataset enhanced with image augmentation has been renamed Weather-Augmented WGISD (WA-WGISD), and its subdivision of images in training/validation and testing is shown in the second row of Table 2.

4.1.2. Training Strategy

The actual training and validation of the neural network for object detection on grape clusters has followed three main lines of implementation, in order to be able to control the improvement in the average precision of the network itself, both globally (i.e., on the whole set of 20 PASCAL VOC classes enriched with the "grapes" class) and at the specific "grapes" class level. The YOLOv2 model has been trained with image pre-processing/resize at 416x416 pixels, on the PASCAL VOC 2007+2012 aggregate dataset and the original WGISD (**416@VOC+WGISD**), then with image pre-processing/resize at 416 × 416 pixels, on the PASCAL VOC 2007+2012 aggregate dataset and the Weather-Augmented WGISD (**416@VOC+WA-WGISD**) and finally with image pre-processing/resize at 608 × 608 pixels, on the PASCAL VOC 2007+2012 aggregate dataset and the Weather-Augmented WGISD (**608@VOC+WA-WGISD**).

An example of the graphical monitoring for training the **608@VOC+WA-WGISD** network is shown in Figure 5.

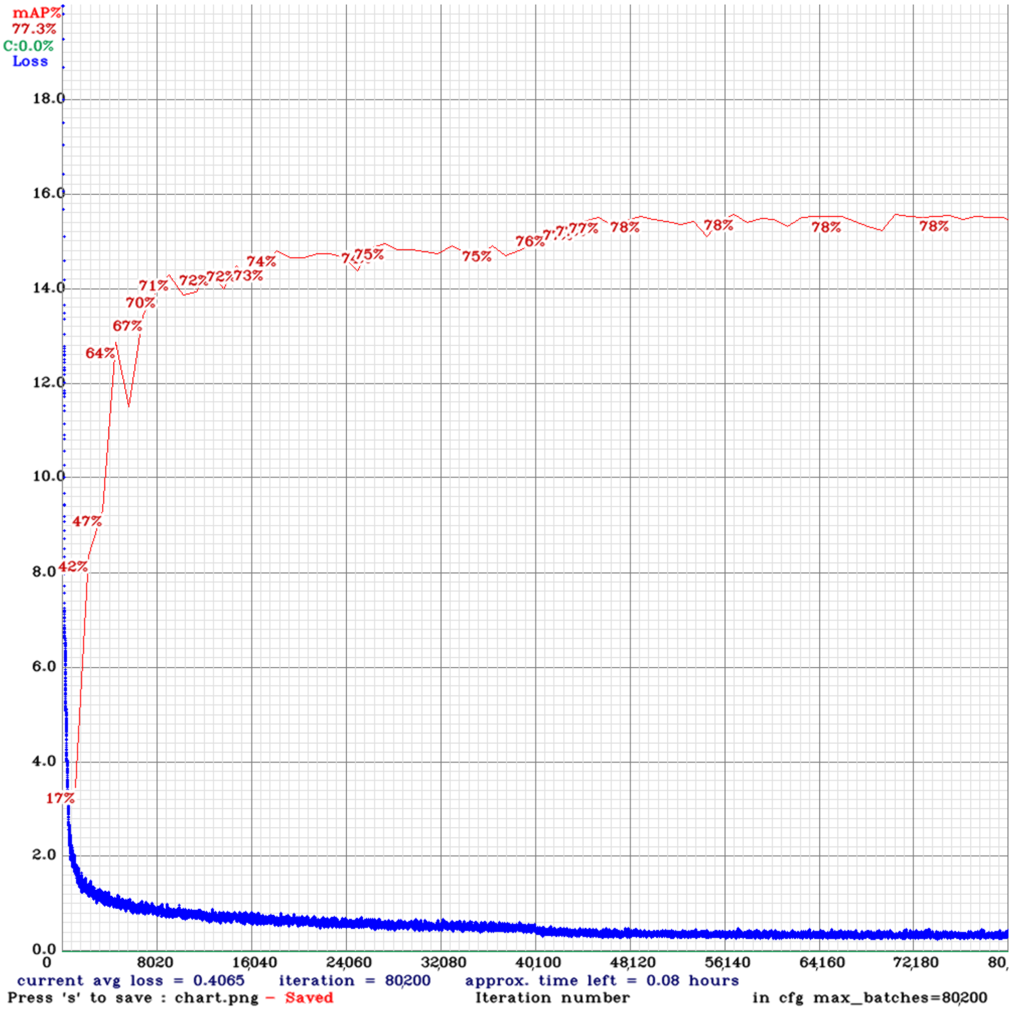


Figure 5. Average precision/loss curve for **608@VOC+WA-WGISD** training.

Each training cycle completed successfully underwent an optimization procedure for OpenCL and translation into OpenCL Caffe standard to be integrated with the developed software components (Section 3.2.2). The main evaluation parameters on the quality of the resulting neural networks are reported in Table 4.

Comparing, in particular, the medium average precision parameters on the entire set of classes (**mAP@.5**) and the average precision on the "grapes" class (**AP@.5**), the adopted choices for augmenting the WGISD dataset and a lower resize in pre-processing have positively influenced the learning curve. All this without exceeding in quality percentages on the accuracy, which could indicate a potential overfitting of the network with respect to the training data.

Table 4. Neural network performance summary for grape recognition.

	416@VOC+ WGISD	**416@VOC+ WA-WGISD**	**608@VOC+ WA-WGISD**
# detections	58,313	94,062	88,623
# unique truths	12,882	12,979	12,979
Precision@.25	0.72	0.67	0.70
Recall@.25	0.72	0.75	0.78
F1-score	0.72	0.71	0.74
TP@.25	9237	9788	10,147
FP@.25	3530	4768	4386
FN@.25	3645	3191	2832
avg IoU	0.55	0.52	0.53
mAP@.5	0.72	0.75	0.77
AP@.5 ("grapes" class)	0.50	0.66	0.80

4.2. *Vision Capabilities Simulations*

The developed networks were subsequently adopted for inference simulations in controlled environments and during various site visits in a vineyard lying on the hills near Treviso, Veneto, Italy (N 45°41′56.066″, E 12°34′15.697″). Since the neural networks' development took place during the late spring months (when a grapevine passes from flowering to budding), the simulations were carried out on faithful plastic replicas of dark red, light red and white fully ripe grape clusters (Figure 6).

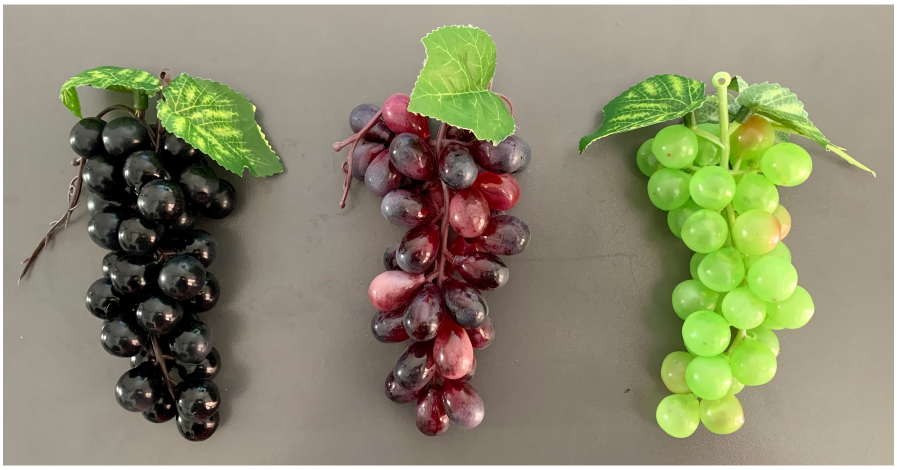


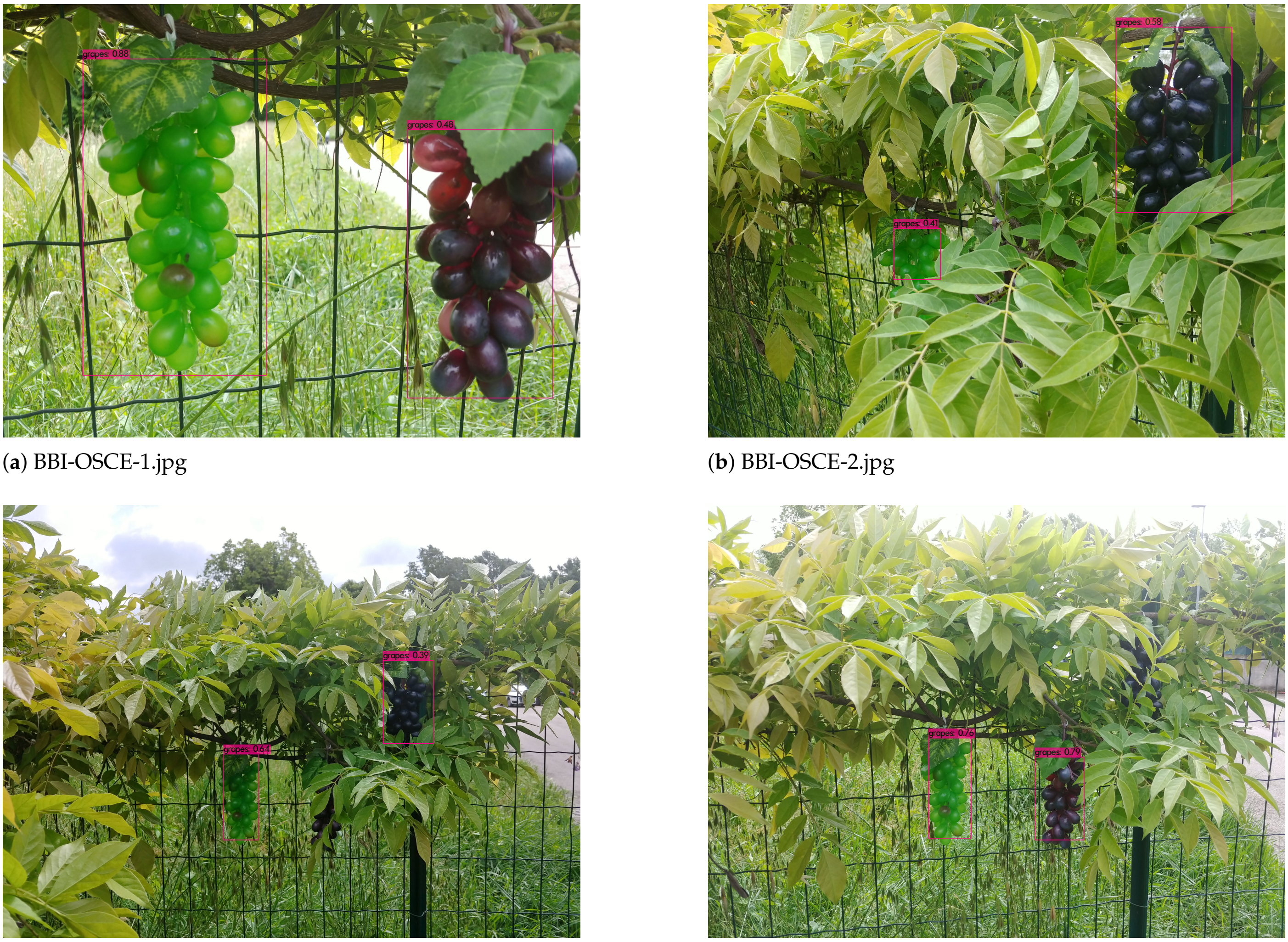
Figure 6. Grape cluster replicas used during simulations.

4.2.1. Recognition in Indoor Controlled Environment

The first recognition experiments have been carried out in a closed-space laboratory in conjunction with the first spraying tests. The environment and recognition results are summarized later in Section 4.3.1. In line with the numeric analysis reported in Table 4, during the laboratory simulations the work team noticed a grape cluster detection success rate clearly in favor of the **608@VOC+WA-WGISD** network, which was taken as a reference network for all subsequent experiments.

4.2.2. Recognition in Open-Space Controlled Environment

Figure 7 reports some of the network's inference results using some photographs taken in an open-space, yet controlled test environment, where grape cluster replicas have been hung next to real foliage.



(**a**) BBI-OSCE-1.jpg

(**b**) BBI-OSCE-2.jpg

(**c**) BBI-OSCE-3.jpg

(**d**) BBI-OSCE-4.jpg

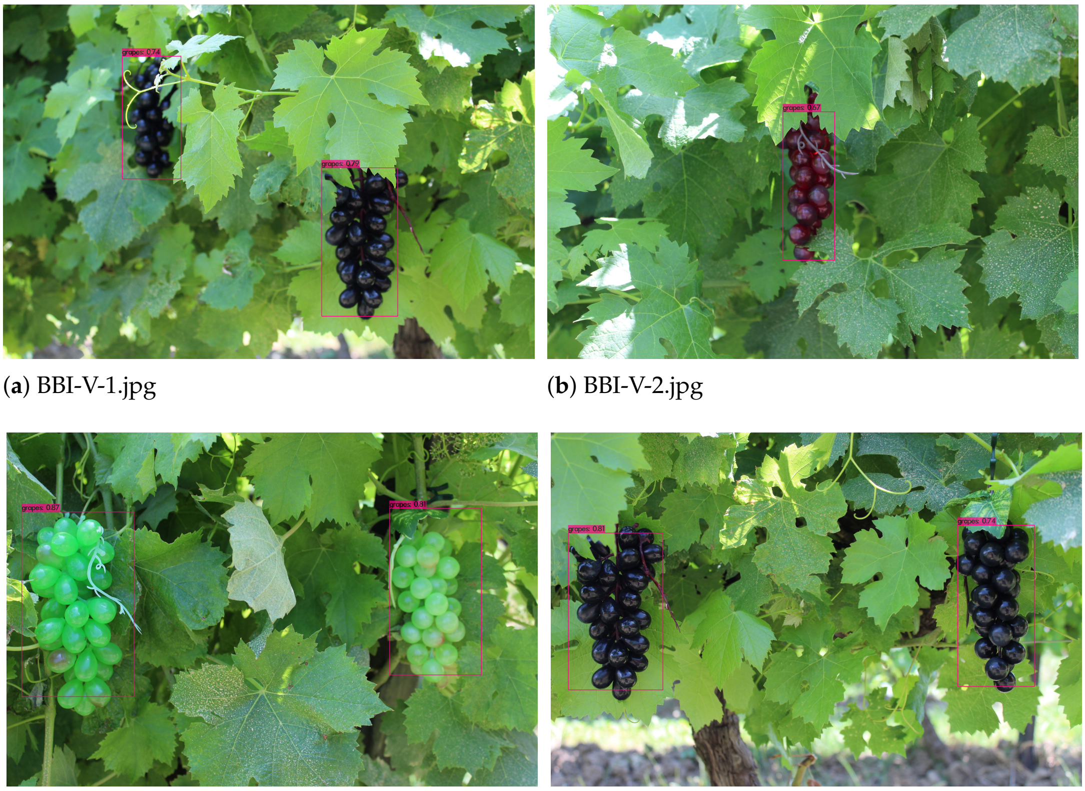
Figure 7. Bounding Box Inference from **608@VOC+WA-WGISD** in Open-Space Controlled Environment (BBI-OSCE).

The bounding boxes shown on each image (labeled with "grapes") were inserted synthetically by the script that applies the inference result and represents the position returned by the network for each recognized cluster replica, with the related confidence percentage.

4.2.3. Recognition in the Vineyard

In this case too, the recognition accuracy of the **608@VOC+WA-WGISD** network was significantly higher than all the other implementations. Figures 8 and 9 show the detection results on some images taken in the vineyard, where the grape cluster replicas were placed on the vine shoots at different heights and degrees of occlusion caused by the foliage.

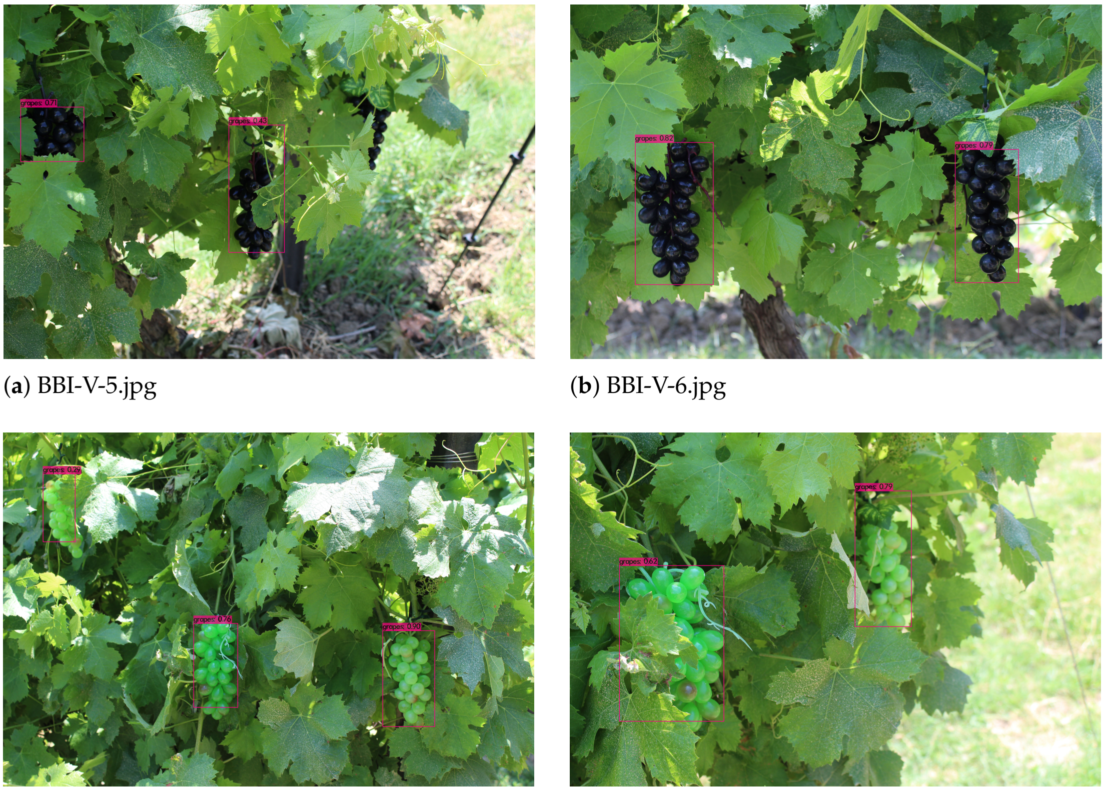
Notice that occlusion in image detection is an open problem for which only experimental work has been performed so far with incomplete solutions. However, even if foliage occlusion is a problem to be dealt with, there is no need to solve the occlusion problem among bunches. In fact, the decision whether to spray or not (and specifically, where) does *not* involve the most difficult task of isolating one bunch from another, as in the case of bunch counting.



(**a**) BBI-V-1.jpg (**b**) BBI-V-2.jpg

(**c**) BBI-V-3.jpg (**d**) BBI-V-4.jpg

Figure 8. Bounding Box Inference of **608@VOC+WA-WGISD** in Vineyard (BBI-V)-Part 1/2.



(**a**) BBI-V-5.jpg (**b**) BBI-V-6.jpg

(**c**) BBI-V-7.jpg (**d**) BBI-V-8.jpg

Figure 9. Bounding Box Inference of **608@VOC+WA-WGISD** in Vineyard (BBI-V)-Part 2/2.

4.2.4. Comparison with State of the Art

In order to have a standardized basis for comparison and to assess the performance of the **608@VOC+WA-WGISD** against the state of the art, two distinct exploratory studies have been conducted.

First, the recognition capabilities of **608@VOC+WA-WGISD** have been validated with two well-known datasets; that is to say, the Grapevine Bunch Detection Dataset version 2 (GBDDv2, https://zenodo.org/records/7717055, accessed on 21 March 2024), and the Open Image Dataset version 7 (OIDv7, https://storage.googleapis.com/openimages/web/visualizer/index.html, accessed on 4 April 2024). Table 5 reports the results against the evaluation and test splits for both datasets.

Table 5. Detection results of **608@VOC+WA-WGISD** over GBDDv2 and OIDv7 (restricted to Grape class), with confidence threshold that optimizes F1-score.

		IoU	Conf. Threshold	Precision	Recall	F1-Score
GBDDv2	Evaluation	0.40	0.05	0.94	0.79	0.86
		0.45	0.05	0.86	0.77	0.81
		0.50	0.05	0.78	0.75	0.77
	Test	0.40	0.05	0.96	0.82	0.89
		0.45	0.05	0.90	0.81	0.85
		0.50	0.05	0.82	0.80	0.81
OIDv7 Grape (All)	Evaluation	0.40	0.05	0.68	0.28	0.39
		0.45	0.05	0.48	0.21	0.29
		0.50	0.05	0.48	0.21	0.29
	Test	0.40	0.05	0.61	0.28	0.39
		0.45	0.05	0.56	0.27	0.36
		0.50	0.05	0.52	0.25	0.34
OIDv7 Grape (Bunches)	Evaluation	0.40	0.05	0.64	0.30	0.40
		0.45	0.05	0.46	0.23	0.31
		0.50	0.05	0.46	0.23	0.31
	Test	0.40	0.05	0.60	0.48	0.53
		0.45	0.05	0.56	0.47	0.51
		0.50	0.05	0.53	0.45	0.49

The choice to conduct several tests with different IoU thresholds is motivated by the different bounding box definition strategy adopted for GBDDv2 and OIDv7 as compared to WA-WGISD. In fact, the first two datasets provide, in general, wider bounding boxes to determine the position of each grapevine bunch in the images, whilst the WA-WGISD applies a more conservative policy (i.e., narrower bounding boxes). For this reason, adopting lower IoU thresholds to classify TPs, FPs and FNs represents a legitimate choice for an unbiased comparison against these datasets.

Being created specifically for grapevine bunch detection, the GBDDv2 evaluation and test sets have been used in their entirety for the evaluation round. In this regard, the numbers reported in the first and second rows of Table 5 show a remarkable recognition behaviour, which far outperforms both the precision and F1-score metrics as calculated against the evaluation and test splits of the VOC+WA-WGISD for the "grapes" class, while keeping the recall almost aligned with the expected outcome.

On the other hand, some clarifications are required to correctly explain the decrease in the detection performances over the OIDv7, which has been preemptively framed to the Grape class (third and forth rows of Table 5). Notice that the **608@VOC+WA-WGISD** has been trained with the purpose of detecting whole grape clusters in the vineyard (bunches) and not single grape berries, whilst the Grape class of the OIDv7 considers bounding boxes for all of the following:

- Single grape berries;

- Grape bunches in environments other than the vineyard (for instance, markets and kitchens);
- Berries and clusters of completely different varieties of fruit, like currants (which can also be considered a counter-measure to mitigate overfitting problems during network training with OID).

This justifies the clear improvement (nearly 20%) in the **608@VOC+WA-WGISD** detection performances against the OIDv7 when the Grape class is pre-processed in a way that single berries are not considered in the counting of TP, FP and FN (fifth and sixth rows of Table 5). It is arguable that not considering clusters and bunches of other fruit could further improve the detection capabilities of the **608@VOC+WA-WGISD** against the dataset.

The rationale behind the evaluation of the **608@VOC+WA-WGISD**'s capabilities on different datasets is crucial to express an objective opinion when comparing the proposed CNN network to recently proposed models for grapevine bunch detection, which was the subject of the second exploratory study, and whose results are reported in Table 6.

Table 6. Comparison between the proposed model and the state-of-the-art DL for grapevine bunch detection, based on accuracy (Acc.), average precision (AP), medium average precision (mAP) and F1-score (results taken from [33]).

State-of-the-Art	Model	Acc. (%)	AP (%)	mAP (%)	F1-Score (%)
Sozzi et al. [34]	YOLOv4	48.90	—	—	—
Aguiar et al. [35]	SSD MobileNet v1	—	—	66.96	—
Sozzi et al. [27]	YOLOv5x	—	—	79.60	—
Santos et al. [26]	Mask R-CNN	—	—	—	84.00
Yin et al. [25]	Mask R-CNN	—	89.53	—	—
Li et al. [36]	YOLO-Grape	—	—	—	90.47
Ghiani et al. [37]	Mask R-CNN	—	—	92.78	—
Pinheiro et al. [33]	YOLOv7	—	—	77.00	94.00
Proposed (best)	YOLOv2	58.43	80.00	77.00	74.00

The metrics regarding the proposed model are quite in line with the state of the art, except from F1-score. Notice, however, only the F1-score of [26] is directly comparable with the performance of the **608@VOC+WA-WGISD**, being the Mask R-CNN in their work trained on a subset (WGISD) of the VOC+WA-WGISD. On the other hand, although Refs. [33,36] show impressive recognition performances, their training processes have been conducted on WGISD-unrelated datasets (specifically, 2331 grape images in grape fields in Shihezi and Turpan, Xinjiang from July 2020 to September 2020 for [36], and the GBDDv2 for [33]); only a comparative evaluation of their models against the VOC+WA-WGISD would give a more faithful meter of comparison. For instance, the results in Table 5 directly compare the **608@VOC+WA-WGISD** with the model in [33], showing a best-case F1-score of 89% against the GBDDv2.

4.3. Cobotic Preliminary Experiments

This section is devoted to the introduction of the experiments conducted as preliminary tests in order to prepare the apparatus for the second experimental phase; that is to say, testing the cobot in the vineyard as a system for chemical distribution.

4.3.1. Spraying Test

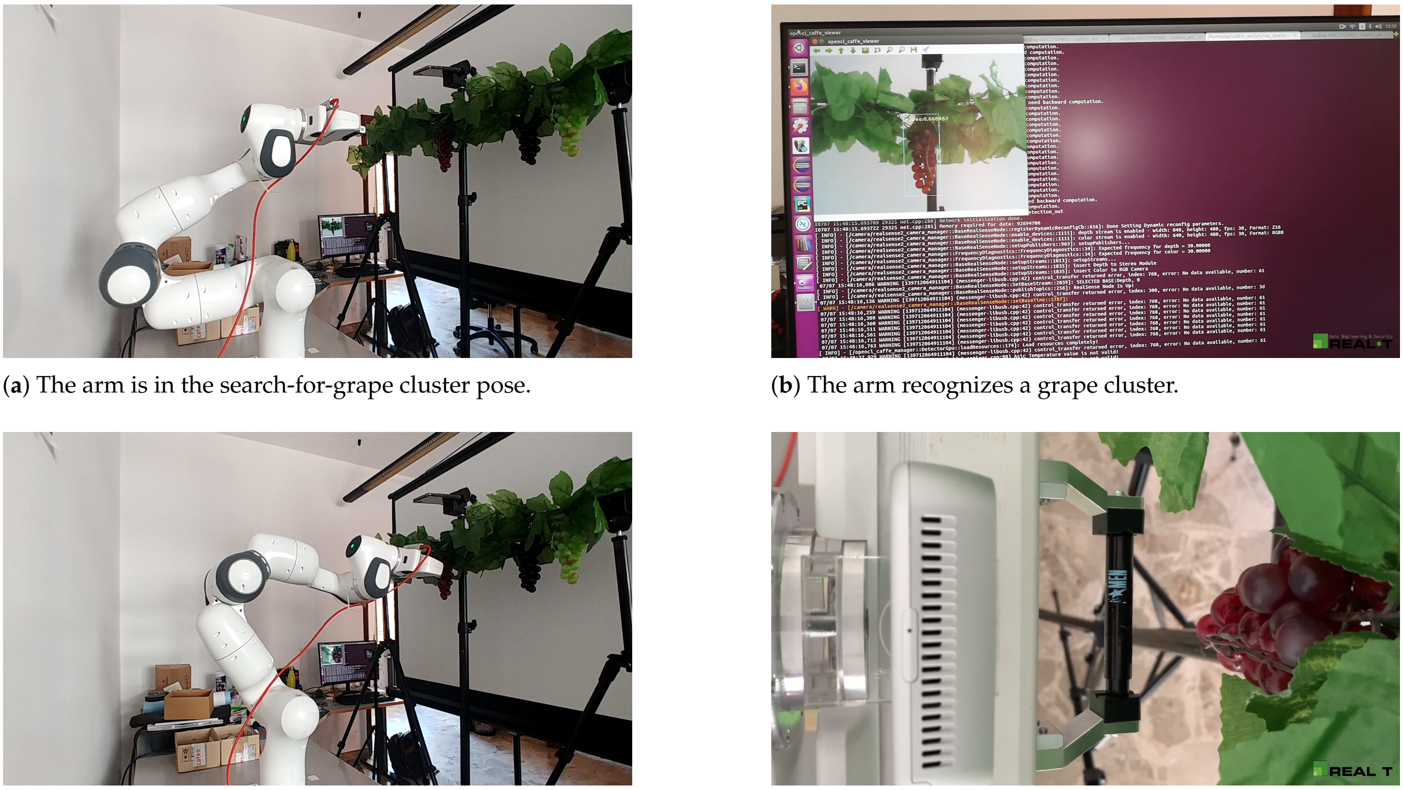
The Franka Hand end effector was adapted to include a sampler of a spray perfume filled with water so that the functionality could be proven without incurring additional research and development costs to purchase professional chemical dispensers as end effectors.

In detail, the spraying process encompasses several phases, from the home pose towards moves for searching (see Figures 10a and 11a), the decision making by the orchestrator component to devise both the cluster to spray and the pose estimation, the spray position (see Figures 10c and 11c), the actual spraying (Figures 10d and 11d) and the return to the home pose.

Spraying Test in Indoor Controlled Environment

The experimental laboratory tests were carried out by recreating a vineyard environment, using branches to hang fake vine leaves and the aforementioned plastic grape cluster replicas.

In the tests performed, the synthetic grape clusters with the highest recognition percentage were the light red ones (82% on average) followed by dark red (77%) and finally white grape clusters, recognized with an average confidence of 66%. This is probably due to the laboratory lighting conditions, as well as the fact that the used dataset actually contains different degrees of ripeness and grape types. It could also depend on the non-typical background colors used in the laboratory tests: the background was, in fact, set by a white cloth, while in the training datasets, the back of the vines is typically made up of other vines, brown earth and green leaves.



(**a**) The arm is in the search-for-grape cluster pose.

(**b**) The arm recognizes a grape cluster.

```
I0707 15:48:15.693709 29325 net.cpp:280] Network initialization done.
I0707 15:48:15.693722 29325 net.cpp:281] Memory required for data: 92694706
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::registerDynamicReconfigCb::616]: Done Setting Dynamic reconfig parameters.
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::enable_devices::1111]: depth stream is enabled - width: 848, height: 480, fps: 30, Format: Z16
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::enable_devices::1111]: color stream is enabled - width: 640, height: 480, fps: 30, Format: RGB8
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setupPublishers::963]: setupPublishers...
[ INFO] - [/camera/realsense2_camera_manager::FrequencyDiagnostics::FrequencyDiagnostics::34]: Expected frequency for depth = 30.00000
[ INFO] - [/camera/realsense2_camera_manager::FrequencyDiagnostics::FrequencyDiagnostics::34]: Expected frequency for color = 30.00000
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setupStreams::1813]: setupStreams...
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setupStreams::1835]: insert Depth to Stereo Module
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setupStreams::1835]: insert Color to RGB Camera
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setBaseStream::2039]: SELECTED BASE:Depth, 0
 07/07 15:48:16,086 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::publishTopics::256]: RealSense Node Is Up!
[ INFO] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setBaseTime::1787]:
 07/07 15:48:16,136 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 300, error: No data available, number: 3d
[ WARN] - [/camera/realsense2_camera_manager::BaseRealSenseNode::setBaseTime::1787]:
 07/07 15:48:16,259 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,309 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,360 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,511 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,561 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,712 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
 07/07 15:48:16,763 WARNING [139712064911104] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available, number: 61
[ INFO] - [/opencl_caffe_manager::DetectorGpu::loadResources::174]: Load resources completely!
 07/07 15:48:27,929 WARNING [139712064911104] (ds5-options.cpp:88) Asic Temperature value is not valid!
```

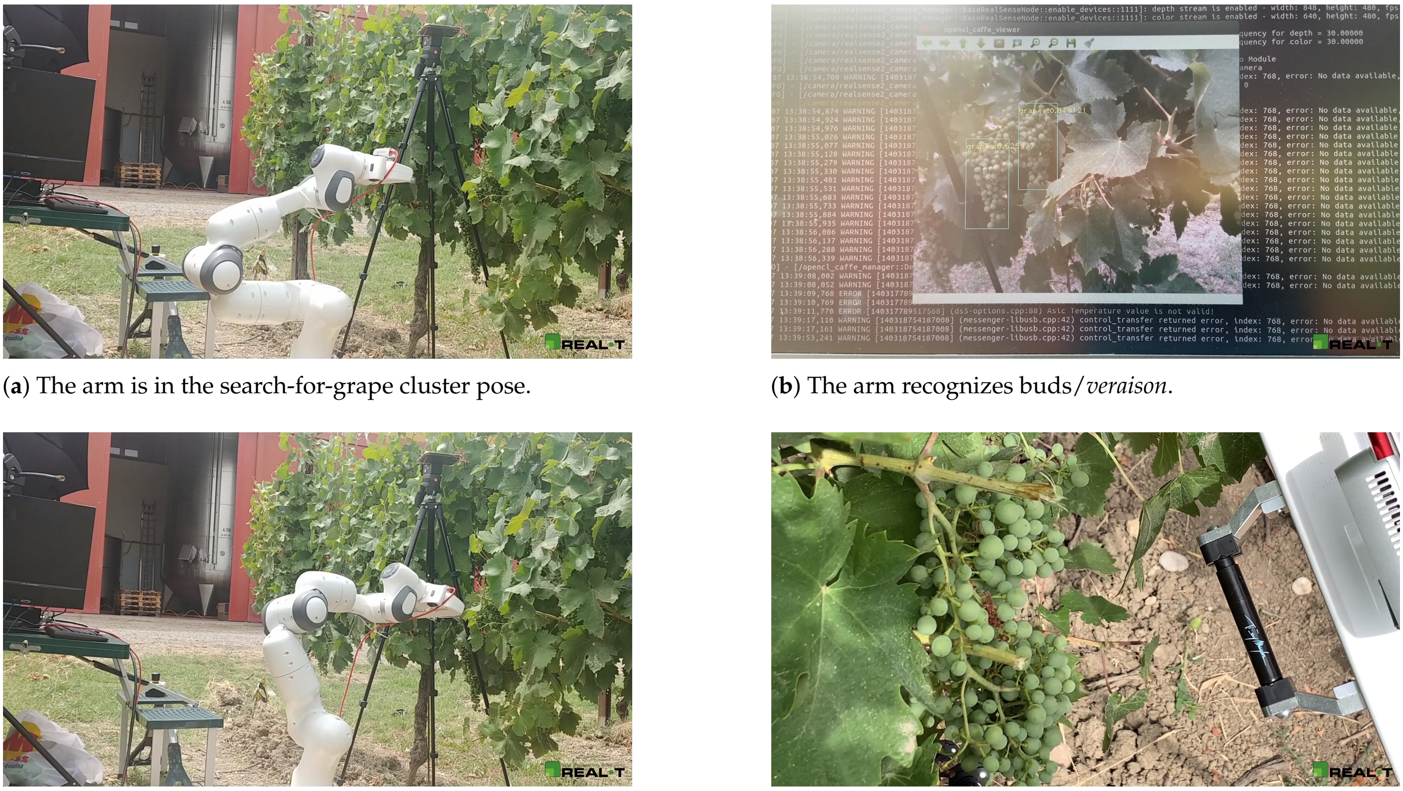
(**c**) The arm approaches a recognized grape cluster.

(**d**) The arm performs spraying action.

Figure 10. Cobotic preliminary experiment: spraying test in indoor controlled environment.

Spraying Test in Vineyard

The developed CNN was able to recognize grape clusters even in a state of buds/*veraison*; that is, the phenological state that precedes the complete ripening of the bunch. It was therefore possible to use the robot directly on these, rather than adopting the plastic grape cluster used in the laboratory. This behaviour is supported by the results obtained in the exploratory study of Section 4.2.4 for the GBDDv2 and the OIDv7. Figure 12 depicts how the **608@VOC+WA-WGISD** behaves on the sample buds/*veraison* images extracted from the GBDDv2 test set. A dedicated numerical analysis over both datasets showed that the performance metrics restricted to just the buds/*veraison* are aligned with the overall scores reported in Table 5. This is the reason why the accuracy percentages in the field are comparable to those observed in the controlled environment. The developed procedure seamlessly operates in the vineyard, since the adverse weather conditions (in particular, the high brightness in that area) did not bring any particular problem to the recognition of the adolescent stage of the bunch (Figure 11).



```
[/camera/realsense2_camera_manager::BaseRealSenseNode::enable_devices::1111]: depth stream is enabled - width: 848, height: 480, fps
[/camera/realsense2_camera_manager::BaseRealSenseNode::enable_devices::1111]: color stream is enabled - width: 640, height: 480, fps
13:39:11,770 ERROR [140317789517568] (ds5-options.cpp:88) Asic Temperature value is not valid!
13:39:17,110 WARNING [140318754187008] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available
13:39:17,161 WARNING [140318754187008] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error: No data available
13:39:53,241 WARNING [140318754187008] (messenger-libusb.cpp:42) control_transfer returned error, index: 768, error
```

(**a**) The arm is in the search-for-grape cluster pose.

(**b**) The arm recognizes buds/*veraison*.

(**c**) The arm approaches to recognized buds/*veraison*.

(**d**) The arm performs spraying action.

Figure 11. Cobotic preliminary experiment: spraying test in vineyard.

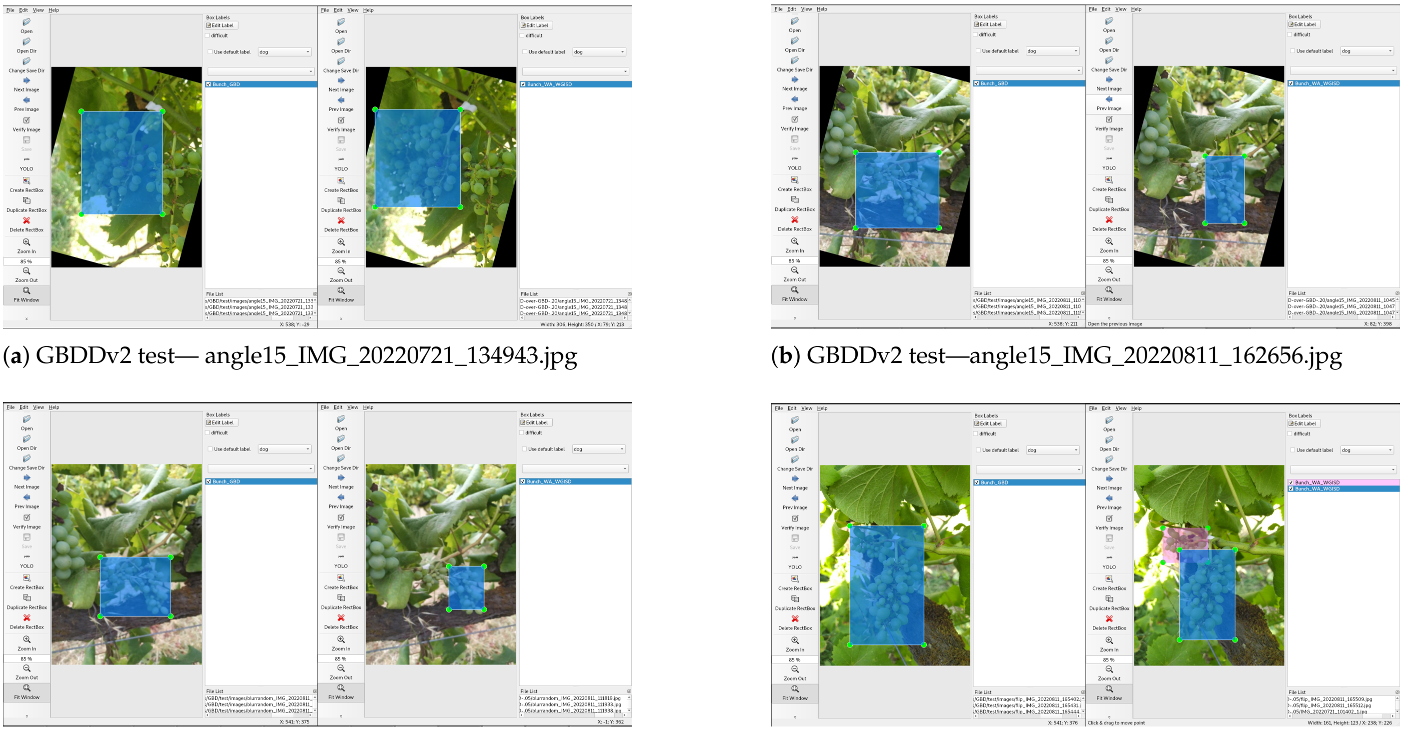
Spraying Performance: Preliminary Considerations

A quantitative study to evaluate the cobotic performances in chemical administration is currently on-going and shall be the subject of a dedicated publication. Nevertheless, here follows some preliminary observations and guidelines to get a glimpse of the rationale in the matter.

The spraying workflow described at the beginning of Section 4.3.1 is relatively straightforward. Once the vision system recognizes a bunch to be treated, the corresponding bounding box is sent to the robotic orchestrator, which, in turn, determines the target pose of the arm based on the *centroid* of the bounding box itself. Based on this premise, it seems legitimate to assume that there is a direct correlation between the performances of the vision system and the performance of the actual administration:

- The *accuracy* of the spraying action, intended as how reliable is the administration process, is directly proportional to the percentage of true positives in the grape recognition process, thus, it is very likely interconnected to the concept of *precision* in grape cluster recognition;
- The *efficacy* of the spraying action, deemed as a *completeness* measure, is proportional to the percentage of false negatives in the grape recognition process (i.e., the orchestrator misses a chemical treatment pose because no bounding box has been given), thus, it is strictly related to the *recall* performance metric in grape detection;
- The *efficiency* is measured in terms of the chemical waste during the spraying action. Three main factors play a crucial role in this sense: (i) the percentage of false positives in the grape recognition process (i.e., the vision system feeds the orchestrator with a wrong place to treat chemically); (ii) the precision (in terms of the IoU threshold) of

the detected bounding boxes (i.e., how much the detected bounding box matches the optimal area to treat); (iii) the diffusion strategy that optimizes the extracted bounding box area, given the bounding box width and height, and the technical specifications of the chemicals diffuser (e.g., diffusion shape, diffusion distance, dispersion rate and so on).



(**a**) GBDDv2 test— angle15_IMG_20220721_134943.jpg (**b**) GBDDv2 test—angle15_IMG_20220811_162656.jpg

(**c**) GBDDv2 test—blurrandom_IMG_20220811_162656.jpg (**d**) GBDDv2 test—IMG_20220721_103744_1.jpg

Figure 12. Examples of buds/*veraison* bunch detection of **608@VOC+WA-WGISD** for GBDDv2 test images (right in each image), compared with corresponding ground truth (left in each image).

5. Conclusions

In this paper, a collaborative robotic, vision-aided solution for automatic chemical treatment on grape clusters has been described. The overall architecture has been detailed, as well as the principal components to be used, the neural training and the test phase. The experimental results with respect to the automatic detection for both grape clusters and the zone of precision spraying have been given, showing it can be feasible in a real-world scenario using a cobotic arm under a low-cost effective constraint.

Many integrations, open issues and enhancements are currently in the research agenda and shall be matter of subsequent publications as follows:

- An on-going dedicated study of the quantitative evaluation of cobotic chemical spraying performances (related, for instance, to the accuracy and stability of the entire system)—see the last part of Section 4.3.1;
- A dedicated study of autonomous, low-cost-oriented navigation in an open vineyard field;
- A theoretical and applied study on how the vision system and the overall devised architecture for chemical spraying should be modified in order to cope with other crucial tasks in viticulture environments, e.g., mature grape physical harvesting.

Author Contributions: Conceptualization, C.T., A.P. and S.S.; data curation, A.P.; formal analysis, C.T. and S.S.; investigation, C.T., M.C., F.O. and S.S.; methodology, C.T., A.P. and S.S.; project administration, C.T.; software, A.P. and S.S.; supervision, C.T.; validation, S.S.; writing—original draft, A.P., M.C., F.O. and S.S.; writing—review & editing, M.C., F.O., C.T. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author(s). Embrapa WGISD description and training data are freely available at https://github.com/thsant/wgisd, accessed on 22 April 2024. PASCAL VOC 2007 and 2012 descriptions and training data are freely available at http://host.robots.ox.ac.uk/pascal/VOC/, accessed on 22 April 2024. Grapevine Bunch Detection Dataset version 2 descriptions and training data are freely available at https://zenodo.org/records/7717055, accessed on 21 March 2024. Open Image Dataset version 7 descriptions and training data are freely available at https://storage.googleapis.com/openimages/web/visualizer/index.html, accessed on 4 April 2024. All rights are reserved for images reported in the present work, unless explicitly specified.

Acknowledgments: Authors gratefully thank Antonino Parisi for his valuable help in the revision process. His hints and suggestions provided appreciable help that improved the paper in a significant manner.

Conflicts of Interest: The authors declare the following financial interests/personal relationships that may be considered as potential competing interests: Claudio Tomazzoli reports a relationship with Real T S.R.L. that includes board membership and equity or stocks. All other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

PA	Precision Agriculture
CPS	Cyber–Physical System
ICT	Information and Communication Technology
SME	Small–Medium Enterprise
CV	Computer Vision
RFID	Radio Frequency Identification
CNN	Convolutional Neural Network
(R)OS	(Robot) Operating System
MRTP	Mobile Robot Programming Toolkit
YARP	Yet Another Robot Platform
SLAM	Stop, Look, Ask, and Manage
CPU	Central Processing Unit
RAM	Random Access Memory
NIC	Network Interface Card
LAN	Local Area Network
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
YOLO	You Only Look Once
WGISD	Wine Grape Instance Segmentation Dataset
WA-WGISD	Weather-Augmented WGISD
PASCAL	Pattern Analysis, Statistical Modelling and Computational Learning
VOC	Visual Object Classes
TP	True Positive
FP	False Positive
FN	False Negative
IoU	Intersection over Union
AP	Average Precision
mAP	Medium Average Precision
BBI-OSCE	Bounding Box Inference-Open-Space Controlled Environment
BBI-V	Bounding Box Inference-Vineyard

References

1. Lee, J.; Bagheri, B.; Kao, H.A. A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems. *Manuf. Lett.* **2015**, *3*, 18–23. [CrossRef]
2. Jin, Y.; Liu, J.; Xu, Z.; Yuan, S.; Li, P.; Wang, J. Development Status and Trend of Agricultural Robot Technology. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 1–19. [CrossRef]
3. Blackmore, B. A Systems View of Agricultural Robots. In Proceedings of the 4th International Symposium on Intelligent Information Technology in Agriculture, ISIITA 2007, Beijing, China, 26–29 October 2007; pp. 23–31.
4. Ravankar, A.; A. Ravankar, A.; Watanabe, M.; Hoshino, Y.; Rawankar, A. Development of a Low-Cost Semantic Monitoring System for Vineyards Using Autonomous Robots. *Agriculture* **2020**, *10*, 182. [CrossRef]
5. Pretty, J.; Bharucha, Z. Integrated pest management for sustainable intensification of agriculture in Asia and Africa. *Insects* **2015**, *6*, 152–182. [CrossRef]
6. Partel, V.; Charan Kakarla, S.; Ampatzidis, Y. Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Comput. Electron. Agric.* **2019**, *157*, 339–350. [CrossRef]
7. Chang, Y.; Rehman, T. *Current and Future Applications of Cost-Effective Smart Cameras in Agriculture*; CRC Press: Boca Raton, FL, USA, 2017; pp. 75–120. [CrossRef]
8. Wang, A.; Zhang, W.; Wei, X. A review on weed detection using ground-based machine vision and image processing techniques. *Comput. Electron. Agric.* **2019**, *158*, 226–240. [CrossRef]
9. Rehman, T.; Mahmud, M.; Chang, Y.; Jin, J.; Shin, J. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Comput. Electron. Agric.* **2019**, *156*, 585–605. [CrossRef]
10. Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of machine vision in agricultural robot navigation: A review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [CrossRef]
11. Su, D.; Kong, H.; Qiao, Y.; Sukkarieh, S. Data Augmentation for Deep Learning Based Semantic Segmentation and Crop-Weed Classification in Agricultural Robotics. *Comput. Electron. Agric.* **2021**, *190*, 106418. [CrossRef]
12. Su, D.; Qiao, Y.; Kong, H.; Sukkarieh, S. Real Time Detection of Inter-Row Ryegrass in Wheat Farms Using Deep Learning. *Biosyst. Eng.* **2021**, *204*, 198–211. [CrossRef]
13. Tu, S.; Xue, Y.; Zheng, C.; Qi, Y.; Wan, H.; Mao, L. Detection of Passion Fruits and Maturity Classification Using Red-Green-Blue Depth Images. *Biosyst. Eng.* **2018**, *175*, 156–167. [CrossRef]
14. Mohammadi, V.; Kheiralipour, K.; Ghasemi-Varnamkhasti, M. Detecting Maturity of Persimmon Fruit Based on Image Processing Technique. *Sci. Hortic.* **2015**, *184*, 123–128. [CrossRef]
15. Thayer, T.C.; Vougioukas, S.; Goldberg, K.; Carpin, S. Multirobot Routing Algorithms for Robots Operating in Vineyards. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1184–1194. [CrossRef]
16. Agostini, A.; Alenyà, G.; Fischbach, A.; Scharr, H.; Wörgötter, F.; Torras, C. A Cognitive Architecture for Automatic Gardening. *Comput. Electron. Agric.* **2017**, *138*, 69–79. [CrossRef]
17. Zaidner, G.; Shapiro, A. A Novel Data Fusion Algorithm for Low-Cost Localisation and Navigation of Autonomous Vineyard Sprayer Robots. *Biosyst. Eng.* **2016**, *146*, 133–148. [CrossRef]
18. Malneršič, A.; Dular, M.; Širok, B.; Oberti, R.; Hočevar, M. Close-Range Air-Assisted Precision Spot-Spraying for Robotic Applications: Aerodynamics and Spray Coverage Analysis. *Biosyst. Eng.* **2016**, *146*, 216–226. [CrossRef]
19. Adamides, G.; Katsanos, C.; Constantinou, I.; Christou, G.; Xenos, M.; Hadzilacos, T.; Edan, Y. Design and Development of a Semi-Autonomous Agricultural Vineyard Sprayer: Human–Robot Interaction Aspects. *J. Field Robot.* **2017**, *34*, 1407–1426. [CrossRef]
20. Pollard, W.L. Position-Controlling Apparatus. U.S. Patent 2,286,571, 16 June 1942.
21. Roselund, H.A. Means for Moving Spray Guns or Other Devices Through Predetermined Paths. U.S. Patent 2,344,108, 14 March 1944.
22. Huang, T. Computer Vision: Evolution and Promise. *CERN Sch. Comput.* **1996**, *19*, 21–25.
23. Font, D.; Tresanchez, M.; Martínez, D.; Moreno, J.; Clotet, E.; Palacín, J. Vineyard Yield Estimation Based on the Analysis of High Resolution Images Obtained with Artificial Illumination at Night. *Sensors* **2015**, *15*, 8284–8301. [CrossRef]
24. Palacios, F.; Diago, M.P.; Melo-Pinto, P.; Tardaguila, J. Early Yield Prediction in Different Grapevine Varieties Using Computer Vision and Machine Learning. *Precis. Agric.* **2022**, *24*, 407–435. [CrossRef]
25. Yin, W.; Wen, H.; Ning, Z.; Ye, J.; Dong, Z.; Luo, L. Fruit Detection and Pose Estimation for Grape Cluster - Harvesting Robot Using Binocular Imagery Based on Deep Neural Networks. *Front. Robot. AI* **2021**, *8*, 626989. [CrossRef]
26. Santos, T.T.; de Souza, L.L.; dos Santos, A.A.; Avila, S. Grape Detection, Segmentation, and Tracking using Deep Neural Networks and Three-Dimensional Association. *Comput. Electron. Agric.* **2020**, *170*, 105247. [CrossRef]
27. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
28. Ding, Y.; Wu, K. A Multi-Task Learning and Knowledge Selection Strategy for Environment-Induced Color-Distorted Image Restoration. *Appl. Sci.* **2024**, *14*, 1836. [CrossRef]
29. Chen, E.; Chen, S.; Ye, T.; Liu, Y. Degradation-Adaptive Neural Network for Jointly Single Image Dehazing and Desnowing. *Front. Comput. Sci.* **2024**, *18*, 182707. [CrossRef]

30. Li, C.; Hu, E.; Zhang, X.; Zhou, H.; Xiong, H.; Liu, Y. Visibility Restoration for Real-World Hazy Images Via Improved Physical Model and Gaussian Total Variation. *Front. Comput. Sci.* **2024**, *18*, 181708. [CrossRef]
31. Liu, Y.; Yan, Z.; Tan, J.; Li, Y. Multi-Purpose Oriented Single Nighttime Image Haze Removal Based on Unified Variational Retinex Model. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1643–1657. [CrossRef]
32. Liu, Y.; Yan, Z.; Chen, S.; Ye, T.; Ren, W.; Chen, E. NightHazeFormer: Single Nighttime Haze Removal Using Prior Query Transformer. In Proceedings of the 31st ACM International Conference on Multimedia, New York, NY, USA, 29 October–3 November 2023; MM '23; pp. 4119–4128. [CrossRef]
33. Pinheiro, I.; Moreira, G.; Queirós da Silva, D.; Magalhães, S.; Valente, A.; Moura Oliveira, P.; Cunha, M.; Santos, F. Deep Learning YOLO-Based Solution for Grape Bunch Detection and Assessment of Biophysical Lesions. *Agronomy* **2023**, *13*, 1120. [CrossRef]
34. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Grape Yield Spatial Variability Assessment using YOLOv4 Object Detection Algorithm. In *Precision Agriculture*; Stafford, J.V., Ed.; Wageningen Academic Publishers: Wageningen, The Netherlands, 2021; pp. 193–198.
35. Aguiar, A.S.; Magalhães, S.A.; dos Santos, F.N.; Castro, L.; Pinho, T.; Valente, J.; Martins, R.; Boaventura-Cunha, J. Grape Bunch Detection at Different Growth Stages Using Deep Learning Quantized Models. *Agronomy* **2021**, *11*, 1890. [CrossRef]
36. Li, H.; Li, C.; Li, G.; Chen, L. A Real-Time Table Grape Detection Method Based on Improved YOLOv4-Tiny Network in Complex Background. *Biosyst. Eng.* **2021**, *212*, 347–359. [CrossRef]
37. Ghiani, L.; Sassu, A.; Palumbo, F.; Mercenaro, L.; Gambella, F. In-Field Automatic Detection of Grape Bunches Under a Totally Uncontrolled Environment. *Sensors* **2021**, *21*, 3908. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.