robotics

MDPI

Article

Imitation Learning from a Single Demonstration Leveraging Vector Quantization for Robotic Harvesting

Antonios Porichis [1,2,*], Myrto Inglezou [1], Nikolaos Kegkeroglou [3], Vishwanathan Mohan [1] and Panagiotis Chatzakos [1]

[1] AI Innovation Centre, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK; mi23878@essex.ac.uk (M.I.); vishwanathan.mohan@essex.ac.uk (V.M.); p.chatzakos@essex.ac.uk (P.C.)
[2] National Structural Integrity Research Centre, Granta Park, Great Abington, Cambridge CB21 6AL, UK
[3] TWI-Hellas, 280 Kifisias Ave., 152 32 Halandri, Greece; nikolaos.kegkeroglou@twi.gr
* Correspondence: ap21186@essex.ac.uk

Abstract: The ability of robots to tackle complex non-repetitive tasks will be key in bringing a new level of automation in agricultural applications still involving labor-intensive, menial, and physically demanding activities due to high cognitive requirements. Harvesting is one such example as it requires a combination of motions which can generally be broken down into a visual servoing and a manipulation phase, with the latter often being straightforward to pre-program. In this work, we focus on the task of fresh mushroom harvesting which is still conducted manually by human pickers due to its high complexity. A key challenge is to enable harvesting with low-cost hardware and mechanical systems, such as soft grippers which present additional challenges compared to their rigid counterparts. We devise an Imitation Learning model pipeline utilizing Vector Quantization to learn quantized embeddings directly from visual inputs. We test this approach in a realistic environment designed based on recordings of human experts harvesting real mushrooms. Our models can control a cartesian robot with a soft, pneumatically actuated gripper to successfully replicate the mushroom outrooting sequence. We achieve 100% success in picking mushrooms among distractors with less than 20 min of data collection comprising a single expert demonstration and auxiliary, non-expert, trajectories. The entire model pipeline requires less than 40 min of training on a single A4000 GPU and approx. 20 ms for inference on a standard laptop GPU.

Keywords: imitation learning; learning by demonstration; vector quantization; mushroom harvesting; visual servoing

Citation: Porichis, A.; Inglezou, M.; Kegkeroglou, N.; Mohan, V.; Chatzakos, P. Imitation Learning from a Single Demonstration Leveraging Vector Quantization for Robotic Harvesting. *Robotics* **2024**, *13*, 98. https://doi.org/10.3390/robotics13070098

Academic Editors: Dimitrios Loukatos, Konstantinos G. Arvanitis and Georgios Papadakis

Received: 9 May 2024
Revised: 21 June 2024
Accepted: 23 June 2024
Published: 30 June 2024

1. Introduction

There is a significant need for robots to intelligently behave in the face of complex, non-repetitive tasks while minimizing the time and effort required to program a robot for a new activity. Reinforcement Learning (RL) techniques have achieved impressive performance in such complex tasks; however, they require the availability of enormous volumes in terms of data samples as well as per-sample reward annotations which are not always feasible to obtain [1].

Imitation Learning (IL), or Learning by Demonstration, has emerged as an alternative that enables agents to learn how to solve complex tasks by observing the interactions of an expert agent with the environment. Such methods can in principle enable new levels of robotic automation allowing robots to carry out tasks involving high variability with respect to critical characteristics, such as object locations, for which it would be impossible to program a procedural routine [2]. IL, therefore, holds significant promise for industrial applications involving menial work that is still being carried out manually. This is particularly important for sectors where the activities carried out are tedious and of low added value or involve significant health and safety risks.

Fresh white button mushroom (*Agaricus bisporus*) harvesting is an excellent example of such a task. Despite the tremendous uptake of robotic automation, particularly in indoor vertical farming systems, fresh mushrooms are currently being harvested almost exclusively by human mushroom pickers due to the task's difficulty; mushrooms can grow in wildly varying positions and the common paradigm is growing them vertically on systems of wide and long shelves with very short distances between each other. This poses significant challenges for perception. At the same time, however, mushrooms are extremely sensitive to applied forces and can easily blemish which results in poorer quality and revenue loss for the growers. Thus, human harvesters employ motion patterns combining twisting the mushroom before pulling to achieve outrooting with minimal squeezing force [3]. Large mushroom growers have made huge improvements in terms of mushroom yields/m^2 of compost [4], yet they face steep labor shortages that render harvesting a crucial bottleneck in their production pipeline [5,6]. Thus, enabling the automated harvesting of fresh mushrooms is of paramount importance and conventional robotic approaches to this problem require significant engineering in terms of perception, to accurately localize the mushroom, and control, to ensure that mushroom quality is not compromised.

Accurate perception necessitates the use of equipment such as laser-based scanners for highly accurate 3D reconstruction of the mushroom bed [7,8]. Such equipment can be expensive to deploy in an industrial setting and requires thorough calibration at regular intervals. On the other hand, precise and compliant control can be very difficult to accomplish with conventional rigid grippers. Leveraging soft, pneumatically actuated grippers, such as the one shown in Figure 1, is an exceptionally cost-effective solution to the manipulation challenges.

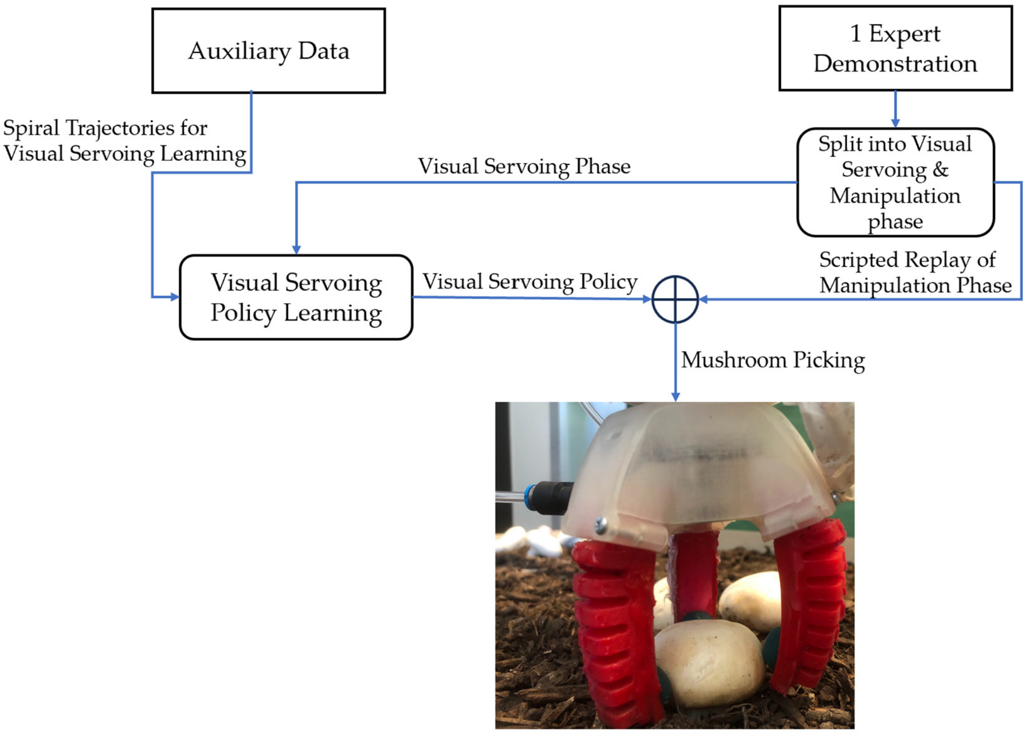

Auxiliary Data
1 Expert Demonstration
Spiral Trajectories for Visual Servoing Learning
Visual Servoing Phase
Split into Visual Servoing & Manipulation phase
Visual Servoing Policy Learning
Visual Servoing Policy
Scripted Replay of Manipulation Phase
Mushroom Picking

Figure 1. Overview of our Imitation Learning approach for mushroom picking with a soft, pneumatic-driven gripper.

Such gripping systems offer an array of advantages besides compliance; they can be inherently recyclable and significantly less expensive than their rigid counterparts, allowing for mass deployment across different harvesting applications. Scaling up the adoption of such novel, low-cost systems would be pivotal in making automation viable for smaller farms which are struggling amidst growing labor shortage pressures [9]. This, however, hinges upon streamlining the perception and control of such systems, which is significantly more challenging due to the lack of proprioceptive sensing, i.e., the absence of measurement of finger position and orientation, on the fingers. Another important challenge is that of hysteresis phenomena, where after several cycles of use, the resting positions of the fingers change over time due to the soft material distortion.

Our approach attempts to tackle the challenges of perception and control for mushroom picking using a soft gripper through an IL technique operating on input from a single RGB camera. An IL pipeline is trained using a single demonstration of a successful trajectory for grasping, twisting, and pulling a mushroom among a cluster. Reaching the mushroom is accomplished by learning a visual servoing controller based on a small set of non-expert trajectories. The main contributions of our work are summarized below:

1. We implement a one-shot IL agent that is capable of mushroom harvesting using a straightforward sensorial stream that is cheap to obtain. The agent operates directly on RGB images from a single camera embedded within the palm of the gripper with no pre-processing, other than downscaling, and the position coordinates of the cartesian robot carrying the gripper. Our method requires no 3D information or camera calibration. No engineering work needs to be carried out apart from collecting ~20 mins worth of data.
2. We introduce a Vector Quantization module that is shown to provide significant performance improvement in terms of mushroom picking success rates. We benchmark against [10], a one-shot IL method that was tested on toy tasks in highly controlled conditions without distractors. Our method is sufficiently robust to achieve a 100% success rate in mushroom picking in a realistic environment in the presence of distractors.
3. We test our approach on a real robot, featuring a soft, pneumatically actuated gripper, shown in Figure 1, and mushrooms of varying sizes. To the best of our knowledge, this is the first implementation of a one-shot IL pipeline on a real setup for mushroom harvesting. This is in contrast to [11], where the IL pipeline required 80 demonstrations and it was only tested in simulated environments.

The rest of the paper is structured as follows: In Section 2, we outline related work with similar aspects to ours in terms of scope. Section 3 details our IL approach for solving the mushroom harvesting task with the soft gripper under the presence of distractors, while Section 4 presents an analysis of the results of our experiments including comparisons with state-of-the-art models used as benchmarks. In the last section, we discuss the results of our work and lay out possible avenues to pursue future research.

2. Background and Related Work

A common family of IL approaches is that of Inverse Reinforcement Learning (IRL) [12], which involves a two-stage process; first, expert agent demonstrations are used to learn a reward function that best explains the trajectories followed, and then an RL algorithm is implemented using the learned reward. One of the first works to demonstrate an IRL method capable of operating on pixel-based input, rather than state vectors, was [13]. This work utilizes a guided cost learning algorithm which operates on visual features which include pruned feature point representations and thus are task-specific. Another pixel-based IRL has been implemented in [14], that leverages a separate neural net to detect key points on the robot configuration. Such methods require a large number of samples to learn during the RL stage. Optimal Transport techniques, pertaining to devising an optimal way of transforming one probability distribution, that of the agent's actions, to another one, i.e., that of the expert's actions, under a cost criterion have been leveraged to mitigate this issue [15]. However, such methods have been demonstrated in simple manipulation tasks that do not require grasping. A novel reward learning scheme based on time contrastive networks is proposed in [16], with the reward computed based on the distance between embeddings of expert and agent observations at similar time points.

Adversarial Imitation Learning (AIL), introduced in [17], is an alternative class of methods that fuse IRL and Generative Adversarial Networks (GANs) [18]. In this regime, after being trained to differentiate between samples obtained from agent and expert trajectories, the discriminator network of a GAN is employed as a reward function in an RL process. Using the Weisserstein distance, a crucial component of GANs, in its primal formulation, techniques like Primal Weisserstein Imitation Learning [19] have proved successful in robotic manipulation straight from pixel-level input on a simulated door opening task. However, this strategy needs about a million steps of interactions to achieve satisfactory performance, which is several orders of magnitude more than our methodology.

A key limitation of IRL and AIL lies in their dependence on an appropriately tuned RL process to train the agent that acts as the final robot controller. This training requires the robot to interact with the environment while still not acting in an optimal and/or safe way. Behavioral Cloning (BC), originally presented in [20], sidesteps this constraint by directly casting IL as a straightforward supervised learning problem, i.e., predicting an action $a = f(o)$, where o is an observation returned by the environment and f is any machine learning model regressing a on o. A notorious issue with BC is that of distributional shift; supervised learning requires that the samples of the training set are independent and identically distributed (i.i.d.), and this is far from the case in BC since past observation–action pairs directly influence the present and future ones. This means that potential discrepancies in the learned policy from the expert one lead the agent to increasingly more foreign states compared to those encountered during training and thus there is a compounding error effect. Nevertheless, BC has been shown to perform remarkably well in practice. An end-to-end visual-based approach that can solve five different manipulation tasks using BC is implemented in [21]. However, this requires a carefully tuned combination of four different neural networks and it relies on >900 demonstrations for certain tasks. A pivotal work in the area of BC is [22], which proposed Implicit Behavioral Cloning (IBC), where actions are predicted by minimizing an energy function defined over pairs of actions and observations. IBC shows significant performance improvements in tasks where optimal trajectories can have multi-modal distributions; however, pixel-based agents have only been tested on a block sorting task with a reduced, 2D action space. The Trajectory Transformer [23], the Behavior Transformer [24], and the Action Chunking Transformer [25] capitalize on the strengths shown by transformer models in capturing relationships over long distances within sequences. Such approaches treat BC as a sequence modeling problem of predicting actions conditioned on observations. Finally, the Perceiver–Actor architecture [26] successfully completes manipulation tasks based on language conditioning but it requires a set of registered RGB-D cameras, and it operates on a 3D voxel-based version of the scene which is computationally heavy to construct. Several recent approaches to BC leverage Denoising Diffusion Probabilistic Models [27], drawing inspiration from their recent successes in modeling high-dimensional data including images and videos. These are generative models that map Gaussian noise to some target distributions, usually conditioned on some context-specific embeddings. Such an approach is presented in [28], implementing a BC pipeline by using a diffusion-based generative model that is conditioned on the current observation embedding and maps Gaussian noise to the next action. The approach has only been tested in simulated environments and video games. A seminal IL approach is Diffusion Policy, introduced in [29], a complex pipeline combining Feature-wise Linear Modulation [30] with a diffusion model, where instead of predicting actions directly, it predicts the gradient field of action energy scores that is then used to obtain a series of actions. This results in a powerful model which, however, comes at a great computational cost, involving over 250 M parameters and requiring days to train on state-of-the art GPUs while requiring a dual-camera setup. Due to the iterative nature of diffusion models, these approaches are much slower and require a lot of empirical tweaks for them to be deployed on real systems.

A common limitation of BC is the requirement of a significant number of expert demonstrations, usually in the order of tens or hundreds of trajectories, to learn sufficiently reliable policies. This can be impractical in cases where kinesthetic teaching, i.e., teaching the robotic manipulator by physically guiding it through the desired motions or tasks, is not feasible as in our case. Our work builds on the findings of [10], which attempt to derive an IL approach that imitates the visual servoing part which is essential in almost all IL methods pertaining to robotic manipulation, with the object handling part of the trajectory following a scripted policy, i.e., by replaying the original expert demonstration's velocities from the grasping point onwards. Indeed, a vast class of tasks can be broken down in a similar manner and harvesting tasks are no exception. An extension of this work attempts to directly learn to estimate the pose of the target object by providing strong inductive biases to the learning approach leveraging geometrical transformations [31]. This approach, like [10], makes a strong assumption about the structure of the scene, namely that the target object to be manipulated is singular and no distractors exist in the camera's field of view. Another approach that loosens this assumption, allowing for the presence of distractors, is that of [32], which employs a sophisticated pipeline that first segments the image to determine the presence of the target object and then processes the image to determine the pose of the target object guiding the visual servoing sequence by trying to match the scale of the detected object with that of the reference trajectory. Although this accomplishes one-shot learning without the use of auxiliary data, it is not practical in our application as we are dealing with scenes where distractors are almost the same as the target object and the target object can vary in scale.

To increase our method's success rate, we make use of Vector Quantization (VQ). Leveraging VQ on observations, rather than actions, has been explored in a limited number of works. VQ is used in [33] to obtain object-centric representations, but the result is processed on a semantic level and the approach focuses on 2D video game playing and self-driving simulation. Another notable work is [34], where VQ is used in a hierarchical IL approach, by quantizing subgoals again in 2D game playing. To the best of our knowledge, our work is the first to utilize VQ directly for visual observation quantization in a robotic manipulation context.

3. Materials and Methods

Our IL implementations follow three core principles that are essentially mandated by the constraints of operating within an agricultural robotics context: (i) that learning should be accomplished without an extensive number of demonstrations as these are cumbersome and often expensive to collect, (ii) that the sensorial streams enabling IL are easy and cheap to collect and process, and (iii) that the computational load of training and inference should be as low as possible so that the learned policy can be deployed at the edge on hardware with moderate capacity. The latter principle entails a preference towards IL methods that are affordable to train and require limited computational infrastructure to be deployed for real systems control.

3.1. Imitation Learning Architecture

Our IL pipeline comprises four stages; an Image Encoder module that allows for the high-dimensional RGB images input to be cast into a low-dimensional embedding; a Vector Quantization module akin to [35] that quantizes the embedding based on a learnable vector codebook, i.e., a set of a pre-determined number of distinct vectors that is the basis of quantization as described next; a Target Position Decoder that takes as input the quantized embedding, z_t, which is concatenated with the encoder measurement of the z-coordinate of the gripper s_t, yielding h_t, and predicts the next target position for the gripper $\widetilde{p}_{t+1} = g(h_t)$; and an Image Decoder that reconstructs the image input. The quantized embedding is produced as a 2D concatenation of selected vectors of the codebook which is in turn learned by minimizing an appropriate loss as explained in the next paragraphs. The overall architecture is shown in Figure 2 below.

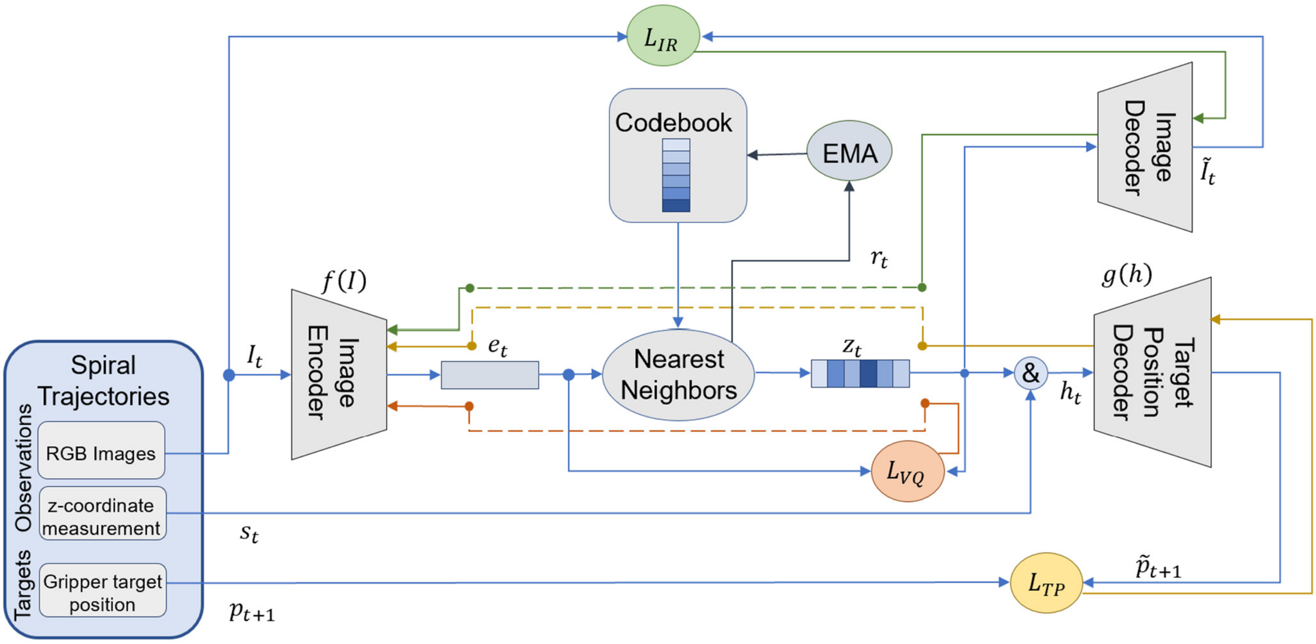


Figure 2. Schematic of Imitation Learning-driven visual servoing approach. The architecture integrates an Image Encoder for processing RGB images, an Image Decoder for frame reconstruction, a Vector Quantization (VQ) module for embedding quantization based on a codebook that is updated using Exponential Moving Average (EMA), and a Target Position Decoder for predicting the Target Position. The system is trained in a self-supervised manner, using a combined loss function that includes reconstruction loss (L_{IR}), VQ loss (L_{VQ}), and the Target Position loss (L_{TP}), facilitating accurate end-effector positioning based on visual and encoder inputs. Dashed lines denote gradient copying to account for the fact that the quantization operation is not differentiable per se.

To train the IL models, we use a dataset, $D = \{T_1 \dots T_M\}$, containing M trajectories. Each trajectory in the dataset, $T_i = \{(o_k,\ p_k),\ k = 1 \dots K_i\}$, is a sequence of observation–action pairs as in the canonical IL setting. In our case, each observation o comprises two elements, namely an RGB image $\boldsymbol{I}$ and the z coordinate of the robot end-effector as obtained by the cartesian robot motor encoders, which is treated as a 1-dimensional vector s_t. Each RGB image $\boldsymbol{I}$ is normalized, with pixel intensities mapped to $[0, 1]$, captured by a camera mounted in the palm of the robot gripper. Thus, we are adopting an eye-in-hand approach.

The Image Encoder module f, parametrized by $\boldsymbol{\theta_f}$, applies a sequence of convolutional residual blocks which downsample each image sample $\boldsymbol{I_t}$ and then flattens the result into an embedding $\boldsymbol{e_t} = f(\boldsymbol{I_t})$. The embedding $\boldsymbol{e_t}$ is passed through the Vector Quantization module q which yields a quantized embedding $\boldsymbol{z_t} = q(\boldsymbol{e_t})$. This quantization process uses a learnable codebook of vectors $C = \{\boldsymbol{b_i},\ i = 1 \dots N\}$ producing z_t as a concatenation of vectors from C in the following manner. Each block of length B of $\boldsymbol{e_t}$ is matched with its nearest neighbor within C in the Euclidean distance sense, leaving a residual $\boldsymbol{r_i}$. The residuals are used to update the codebook using an Exponential Moving Average (EMA) scheme:

$$\boldsymbol{b}_i^{\tau} = \boldsymbol{b}_i^{\tau-1} \times \gamma + \boldsymbol{r_i}(1-\gamma), \tag{1}$$

where γ is an update coefficient between 0.9 and 1. To encourage the Image Encoder to produce embeddings with blocks that are close to the codebook vectors, we implement a commitment loss:

$$L_{VQ}(\boldsymbol{e_t},\ \boldsymbol{z_t}) = (\boldsymbol{e_t} - sg[\boldsymbol{z_t}])^2, \tag{2}$$

where sg[·] denotes stopping the gradient flow to account for the fact that the operation of finding the near neighbor is not differentiable per se. In practice, this means that in computing the gradient of L_{VQ}, z_t is considered independent of the θ_f parameters. The L_{VQ}

loss forces the $\boldsymbol{e_t}$ embedding produced by the Image Encoder to be closer to the quantized embedding produced by the VQ module.

The Target Position Decoder module comprises a lean Long-Short Term Memory (LSTM) Recurrent Neural Network [36] that predicts the target position $\widetilde{p}$. This model is trained to minimize the Mean Squared Error loss:

$$L_{TP}\left(\boldsymbol{p_t}, \widetilde{\boldsymbol{p}}_{\boldsymbol{t}}\right) = \left(\boldsymbol{p_t} - \widetilde{\boldsymbol{p}}_{\boldsymbol{t}}\right)^2, \tag{3}$$

The Image Decoder module is a series of deconvolution layers that outputs a reconstruction $\widetilde{\boldsymbol{I}}$ of the input image.

$$L_{IR}\left(\boldsymbol{I}, \widetilde{\boldsymbol{I}}\right) = \boldsymbol{I}\cdot\log\widetilde{\boldsymbol{I}} + (1 - \boldsymbol{I})\cdot\log\left(1 - \widetilde{\boldsymbol{I}}\right), \tag{4}$$

The training of the entire model proceeds in an end-to-end fashion where all modules are trained jointly by minimizing the aggregate loss:

$$L = \sum\nolimits_{i,t} L_{TP} + \beta L_{VQ} + \lambda L_{IR}, \tag{5}$$

where β is a coefficient modulating the relative weight of the vector quantization loss, also termed commitment loss, and λ is the weight of the Image Reconstruction loss. The summation is performed across all timesteps of all the trajectories.

The gradients from the L_{TP} loss flow back through the Target Position Decoder and the Image Encoder while those of L_{VQ} are only flowing through the latter. The L_{TP} and L_{IR} gradients are side-stepping the discontinuity induced by the quantization module by being copied directly from the Target Position Decoder and the Image Decoder, respectively, to the Image Encoder.

The Target Position Decoder estimates the target position relative to the robot's position in cartesian coordinates. Each such prediction is used to derive a simple velocity controller with a proportional gain K_p. The choice of the controller was driven by simplicity and practicality; we experimentally found that adding integral or derivative terms to the controller offered little benefit in terms of speed in reaching the target position and introduced risks of oscillation around the optimal position. Since the $g(\boldsymbol{h_t})$ estimation is bound to be noisy, we ignore small potential steady-state errors by considering the visual servoing finished when the distance between the current position of the gripper and the target position estimate is less than a threshold δ. The controller equation is thus given as follows:

$$\boldsymbol{v_{t+1}} = \begin{cases} K_p\cdot g(\boldsymbol{h_t}), & |\boldsymbol{m_t} - g(\boldsymbol{h_t})| \geq \delta \\ 0, & |\boldsymbol{m_t} - g(\boldsymbol{h_t})| < \delta \end{cases} \tag{6}$$

In contrast to [11], this end-to-end training approach, illustrated in Figure 2, does not require separate training of the Representation Learning modules. The representational power of the embeddings is allocated in capturing the necessary information to predict the correct target position as well as in reconstructing the original image to keep the embedding manifold as smooth as possible following the lessons of [37].

The model architecture and parameters of the proposed approach are presented in Table 1 below.

Table 1. The proposed model parameters.

Model Component	Type	Layers/Parameters
Image Encoder	Convolutional Neural Network	**Conv layer 1:** 20 channels, 5 × 5, stride 2 **Conv layer 2:** 10 channels, 3 × 3, stride 2

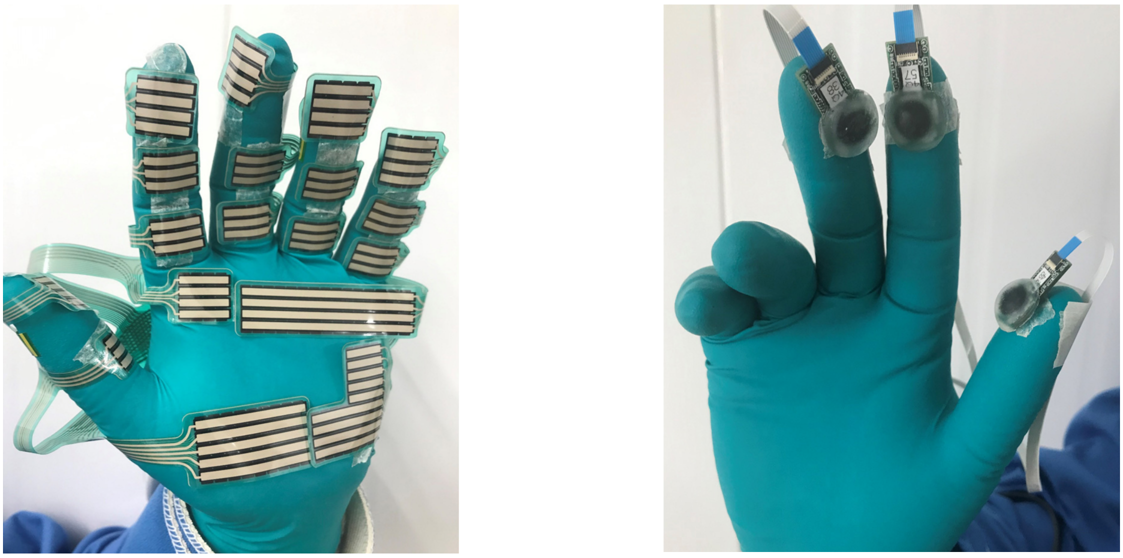
Table 1. *Cont.*

Model Component	Type	Layers/Parameters
Vector Quantizer	EMA-based Quantizer	**Embedding Vocabulary size:** 1024 **Embedding dimension:** 10 **Embedding width/height:** 14 × 21
Target Position Decoder	Recurrent Neural Network	**Sequence length:** 5 **Hidden layer 1:** 1024 **Hidden layer 2:** 1024
Image Decoder	Convolutional Neural Network	**Deconv layer 1:** 20 channels, 3 × 3, stride 2 **Deconv layer 2:** 3 channels, 5 × 5, stride 2

3.2. Experiments

3.2.1. Mushroom Picking Environment

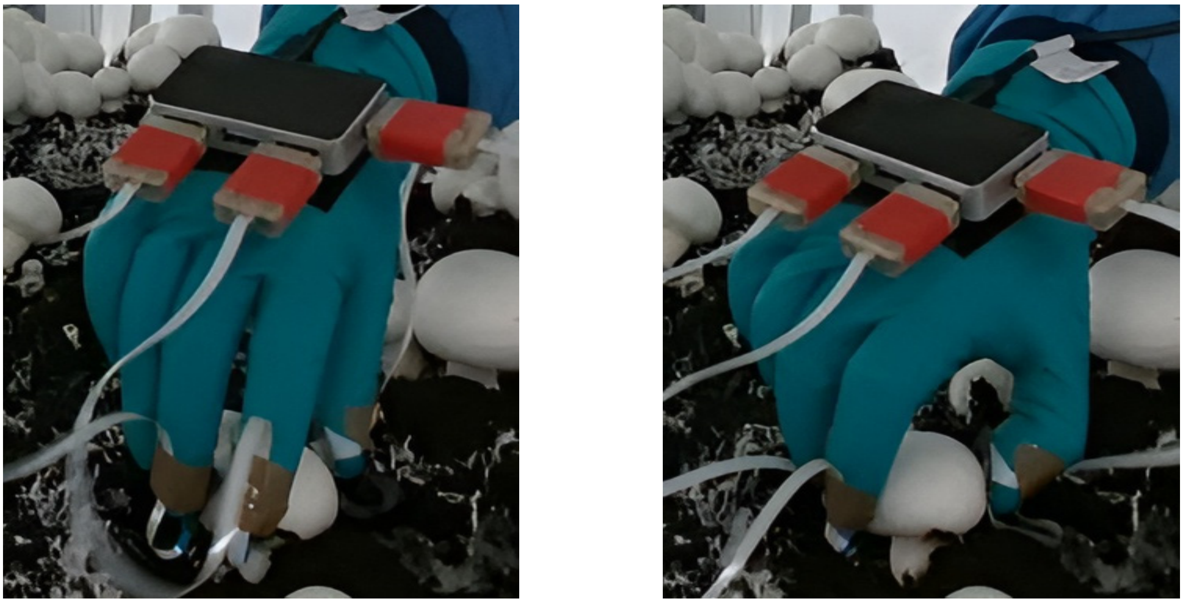
The environment used to test our IL approach was designed to be representative of the actual mushroom harvesting process while staying practical in terms of the equipment and consumables required. The characteristics of the real environment were measured through successive human expert mushroom picking sessions in actual mushroom farms where expert pickers performed harvesting experiments with gloves fitted with pressure and force sensors as seen in Figure 3. Figure 4 shows the two principal steps of outrooting, namely grasping and twisting, as recorded in human expert demonstrations, while Figure 5 shows the pressures and forces measured during a mushroom harvesting demonstration by a human expert.



(**a**) (**b**)

Figure 3. Sensorized gloves for gathering pressure and force data during mushroom picking from human experts. (**a**) Pressure mapping sensors, (**b**) force sensors.

We collected the force and pressure measurements of 30 different picking sequences and manually segmented these into four distinct phases, namely reaching, grasping, outrooting, and lifting. We extracted the maximum force values across the outrooting phases to obtain an estimate of the force limit. In order to synchronize the force and pressure measurements with the camera streams, we recorded the timestamps for each measurement and frame, and we then matched each frame with the closest measurement in time. All sensors and camera frames were captured with the same device to ensure common timestamp baselines.



(**a**) (**b**)

Figure 4. Demonstrations of mushroom harvesting with force sensors mounted on the harvester's gloves: (**a**) grasping the mushroom, (**b**) pulling the mushroom upwards after having twisted it around its principal axis.

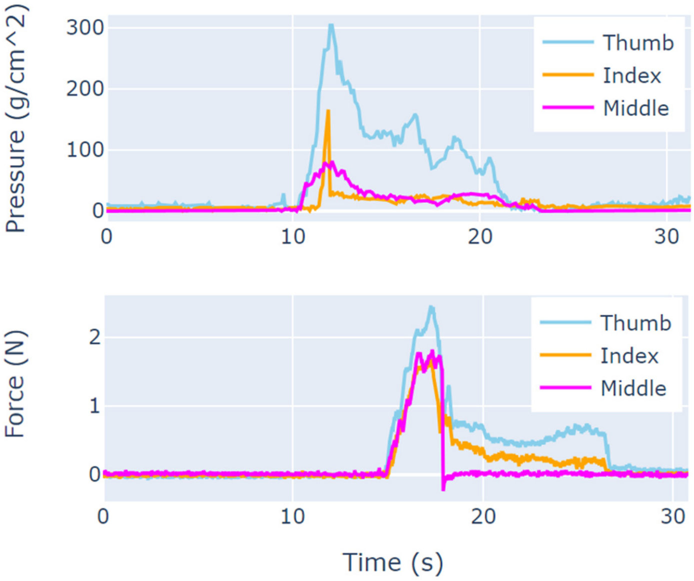


Figure 5. Pressure and force measurements during mushroom harvesting by human experts.

Through these experiments, we established that the expert mushroom pickers keep the forces applied to the mushroom within a range of 2–3 N. Through sperate observations of the twisting motion, in collaboration with the expert picker, we also established that the twisting angle is at least 60°, a finding that is in line with the literature [3]. These findings were used to program the manipulation sequence. This comprises grasping the mushroom by actuating the gripper with 1 bar of pressure, which corresponds to 2.5 N as explained in the next paragraph, then twisting the mushroom by actuating the gripper's *yaw* degree of freedom to rotate, and, finally, moving the gripper upwards along the *z*-axis to lift the mushroom.

3.2.2. Robotic System

The robotic prototype used for the evaluation of the approach is an actuated cartesian robot that has been designed to be compatible with existing mushroom farms. The robot encompasses three actuators, M1, M2, and M3, enabling it to move along a mushroom shelf (x direction) using wheels (M1) and traverse the shelf from side to side (y direction) via a linear slide (M2). An additional linear slide (M3) ensures the ability to lower the robot's end-effector towards the mushroom bed (z-direction). The overall system is shown in Figure 6. A detailed presentation of the robot's configuration and actuation principles is provided in [8].

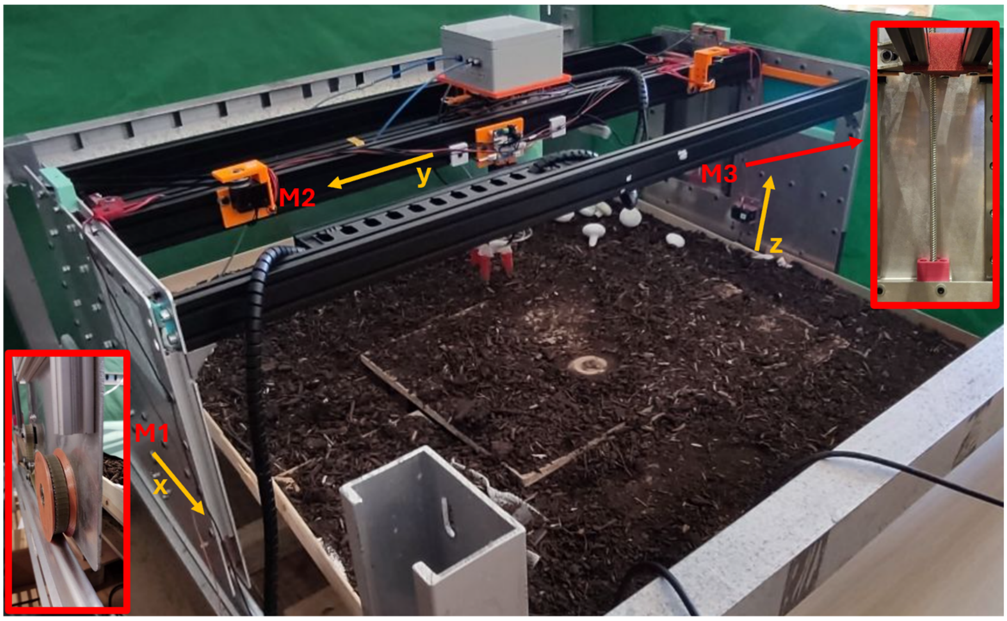


Figure 6. The actuated cartesian robotic system used for mushroom picking trials. The gantry-like robot moves along the x-axis with actuated wheels (M1), and along the y-axis and the z-axis with linear slides (M2 and M3, respectively) [8]. M1 and M3 are shown in inset pictures.

The end-effector of the robot is a soft gripper that has been designed and implemented to replicate the grasping forces recorded during expert demonstration trials. The gripper can deliver a grasping force of max. 2.5 N when 1 bar of pneumatic pressure is applied to inflate the fingers. The gripper's design and operational characteristics are detailed in [38]. The design also allows for integrating an in-hand camera in a straightforward fashion. Figure 7 illustrates the gripper.

The gripper features three actuators, M4, M5, and M6. The first two are used to configure the gripper's angle towards the mushroom bed while the last one is used to deliver the twisting motion. Of the six actuators of the cartesian robot and the gripper, in our experiments, only M1, M2, M3, and M6 were used during the agent trajectory; M4 and M5's angles stayed the same throughout the picking experiments.

For the purposes of visual servoing, within the scope of our work, the gripper motion is controlled with a velocity controller as described in Equation (6). Motion planning is left entirely to the learned agent which calculates the next target position based on visual input from the current image frame.

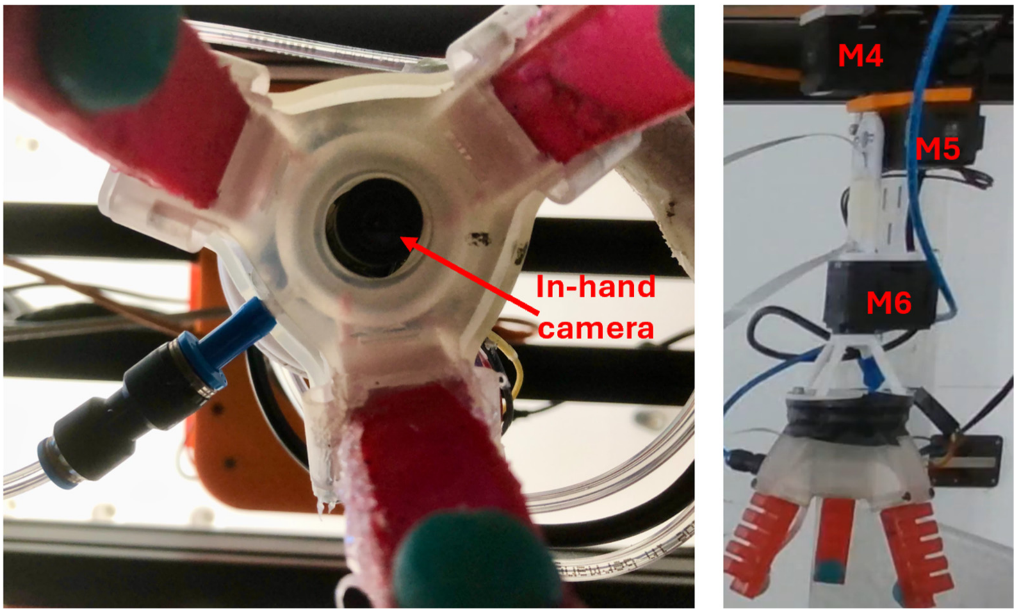


Figure 7. The in-hand camera position as well as the motors (M4, M5, and M6) controlling the motion of the gripper [38].

3.2.3. Data Collection

We follow a simple yet practical approach to collecting the necessary data for learning the visual servoing controller. The procedure can be described as follows:

1. The target mushroom is randomly placed on the soil and a number of other mushrooms are randomly placed around it. All mushrooms are 30–50 mm in cap diameter.
2. The gripper is manually moved in a position that allows for firm grasping, i.e., with the fingers around the target mushroom.
3. The gripper is then moved upwards in a conical spiral with its position and the corresponding image from the in-hand camera recorded at regular intervals. Each observation–relative target position $(o_t,\ p_t)$ is stored. The radius and the slope of the conical spiral are randomized in each data collection.

We collected a total of 17 such trajectories, each with a different mushroom configuration. The total data collection duration was ~20 min and the dataset comprised 16,680 observation–target position pairs. Figure 8 illustrates this process. We also used extensive data augmentation on the images. This involved several key transformations applied to the input images. Firstly, we adjusted the brightness and contrast levels, which aids in simulating varying lighting conditions and improving model generalization. Additionally, we introduce Gaussian noise to emulate real-world sensor noise and enhance the model's ability to handle noisy inputs. Examples of trajectories for data collection are shown in Supplementary Video S1.

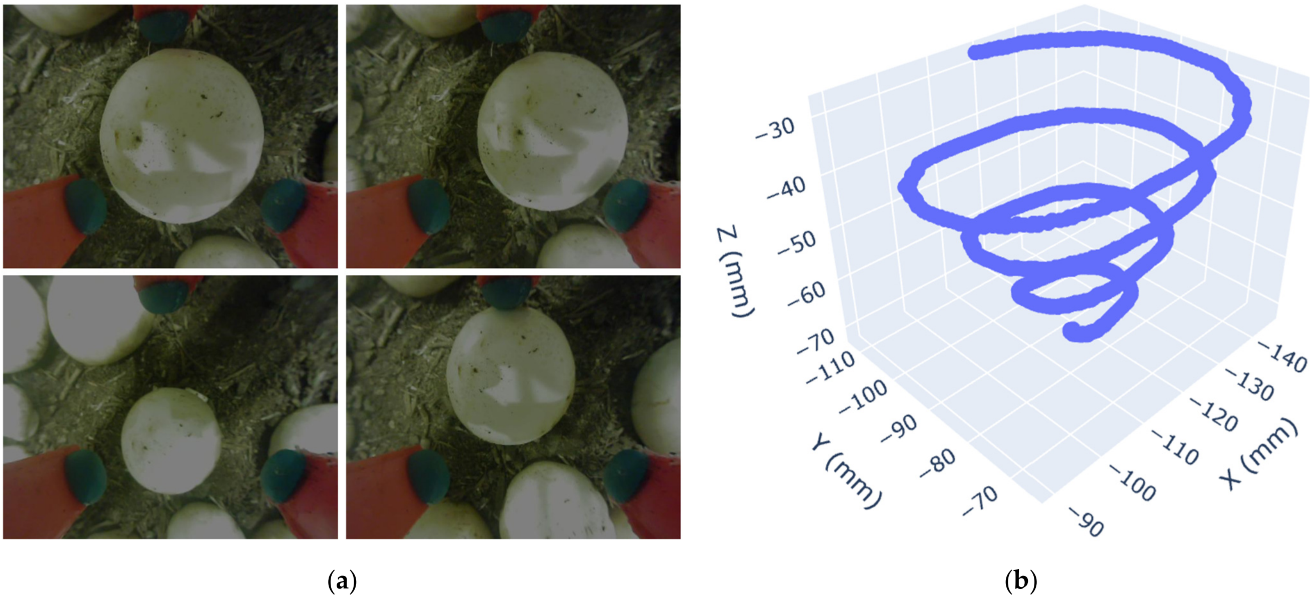


Figure 8. (**a**) Four screenshots of the eye-in-hand camera during a spiral trajectory, (**b**) 3D plot of the trajectory of the gripper in the 3D space.

3.2.4. Benchmarks

Our approach for which we use the identifier, *vq-rec*, is evaluated against the following three benchmarks:

1. A convolutional model following the coarse-to-fine approach of [10] from which our own approach draws inspiration. We refer to this approach as *cnn-c2f*.
2. A simpler variant of our approach where the Image Decoder and the respective loss have been removed to establish the merit of using the Vector Quantization module. This approach is termed *vq-norec*.
3. A non-IL based approach where visual servoing is accomplished leveraging YOLOv5 [39], a well-trusted object detector to detect the mushrooms on the scene, and a controller is programmed to move the gripper to minimize the error between the center of the image and the center of the bounding box of the mushroom closer to the center of the image. This approach is detailed in [8] and we refer to it as *yolo-vs*.

The IL-based approaches, namely *cnn-c2f*, *vq-norec*, and *vq-rec*, are all trained based on the same dataset, collected as described above, and they use a visual servoing controller that moves the gripper to the appropriate position for the grasping, twisting, and lifting motion combination to take place. The latter is replicated directly from a single expert demonstration, similarly to [10]. Separating the visual servoing and the manipulation sequence can be accomplished by simply cutting the trajectory at the point where the yaw angle of the gripper starts changing. This is similar to extracting keyframes like in [26]. The expert demonstration is accomplished by a controller with full knowledge of the optimal position for grasping the mushroom.

The *yolo-vs* controller requires learning just for mushroom detection. This is accomplished by collecting and annotating 200 images of various mushroom configurations.

4. Results

The resulting models were evaluated over 50 trials, each with a cluster comprising between two and five mushrooms freely placed at different positions similar to the data collection process. The learning-based techniques, namely *vq-rec*, *vq-norec*, and *cnn-c2f*, were all trained on the same dataset and the transition from the visual servoing to the scripted twist and lift policy was performed when the prediction of the target position was less than 1 mm, i.e., we choose $\delta = 1$ in Equation (6). Success or failure was determined in a straightforward manner; whenever the gripper still held the mushroom after the lift motion, the episode was deemed successful. The success rate of each of the approaches described above is summarized in Table 2.

Table 2. Success rates in mushroom grasping and lifting of different approaches.

Approach	Vector Quantization	Image Reconstruction	Success Rate
yolo-vs [8]	N/A	N/A	78%
cnn-c2f [10]	No	No	84%
vq-norec	Yes	No	90%
vq-rec	**Yes**	**Yes**	**100%**

As seen in the table above, the proposed approach achieves a remarkable 100% success rate, i.e., it successfully grasped the target mushroom in all instances of the task. Figure 9 illustrates a sequence of frames of a successful episode while further episodes with freely placed mushrooms are provided in Supplementary Video S1. For the rest of the approaches, almost all of the missed cases were mainly due to poor positioning in the z-axis. Indeed, due to the compliant nature of the gripper, a suboptimal alignment in the x-y plane is usually less of a problem as small discrepancies are passively corrected. However, even a few millimeters of deviation from the proper z-position can lead to poor grasping leading to the mushroom slipping off the fingers. The *yolo-vs* approach was shown to be significantly prone to this error mode. We attribute this to the fact that the mushroom scale is not taken into account in such an approach; a larger mushroom seen from a certain height might look very similar to a smaller mushroom observed by a lower height. The IL-based approaches, however, are inherently taking scale into account through the LSTM-based decoder that operates on a sequence of five frames along with the respective z-position of the gripper.

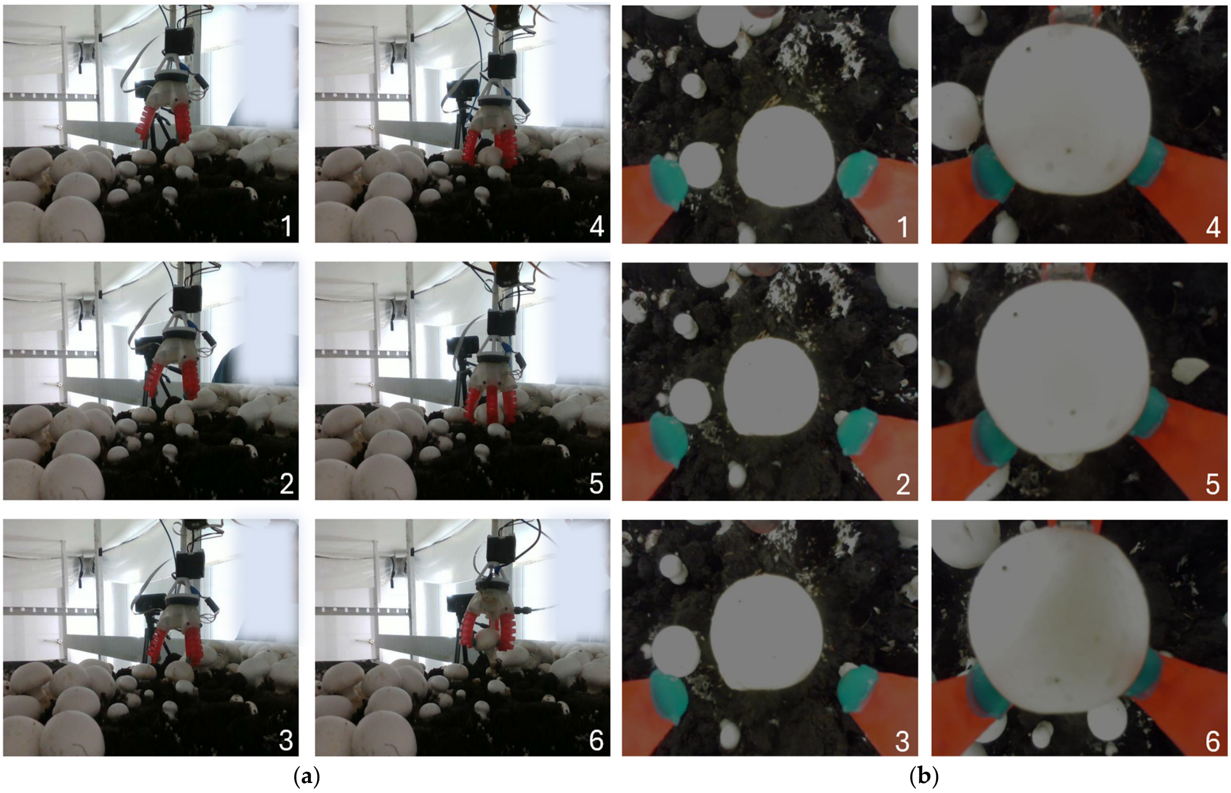


Figure 9. (**a**) Front view and (**b**) eye-in-hand view of a sequence of a successful episode rollout by the trained agent on real mushrooms. (**1**) The gripper starts at a random position; (**2**) moves above the mushroom; (**3**) reaches down; (**4**) grasps; (**5**) twists; and (**6**) pulls the mushroom upwards.

It is worth noting that the introduction of the Vector Quantization module leads to an increase in performance compared to the Convolutional Neural Network model proposed by *cnn-c2f*. Combining Vector Quantization with image reconstruction is shown to significantly improve the accuracy of the predictions. Figure 10 illustrates the predictions produced by the different IL-based agents, based on the observations collected by the expert agent, i.e., a totally scripted agent that has full access to the location of the mushroom to be picked.

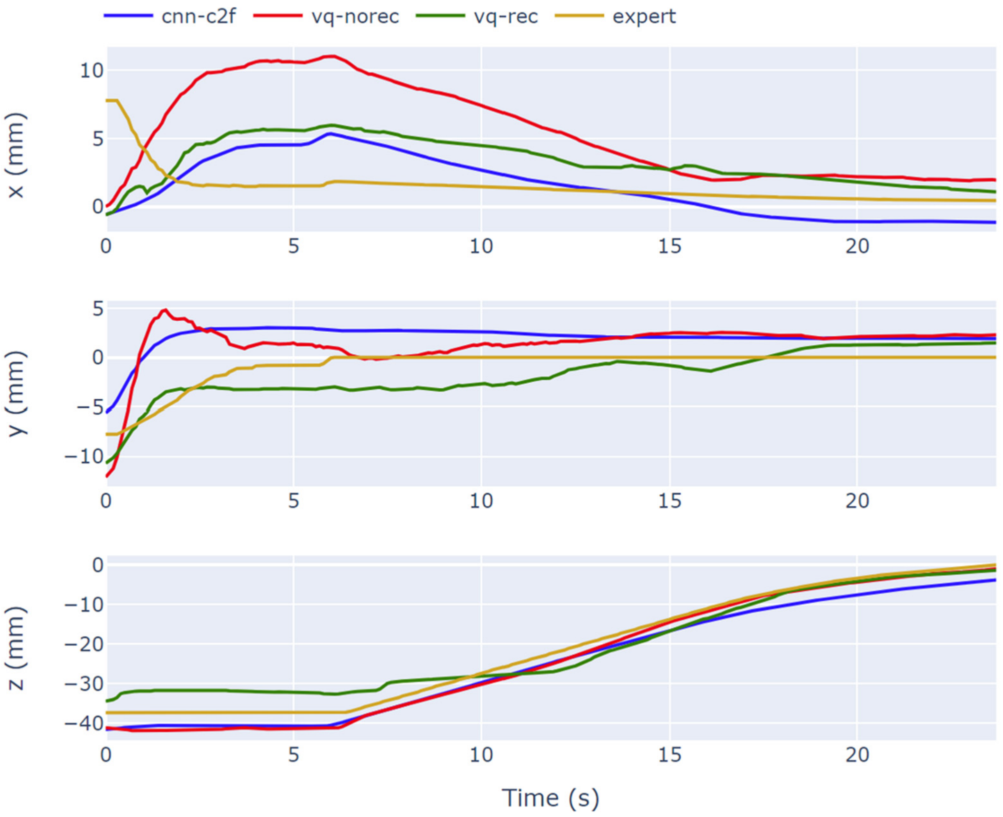


Figure 10. Predictions of different IL pipelines on the expert demonstration. Predictions are produced offline, based on the observations obtained during the expert trajectory.

As seen in Figure 10, the proposed approach combining Vector Quantization and image reconstruction predicts the target position closer to the actual one compared to the rest of the benchmarks, particularly for the time steps closer to the end of the episode. Figure 11 illustrates the trajectories followed by the expert agent and the agent trained with the proposed approach. In contrast to Figure 10, where predictions are produced offline, based on the observations collected by the expert agent, Figure 11 illustrates a full rollout by the learned agent on the same conditions with those of the expert episode, i.e., the same mushroom cluster configuration and the same starting position of the gripper. The proposed agent is able to reach almost precisely the final position of the expert agent, albeit with a delay of ~4 s and with some jittering along the trajectory.

To further investigate the beneficial effects of Vector Quantization and image reconstruction, we conducted an analysis of the embeddings passed as inputs to the Action Decoder across the three different IL-based approaches. To visualize the embeddings, we use the well-trusted t-SNE [40] method to map the high-dimensional vectors onto the 2D plane. Figure 12 illustrates a visualization of the embeddings of each of the models for a single episode of 305 steps. To accentuate the differences between the approaches, we use a relatively low perplexity value of $p = 5$ in line with the low number of samples.

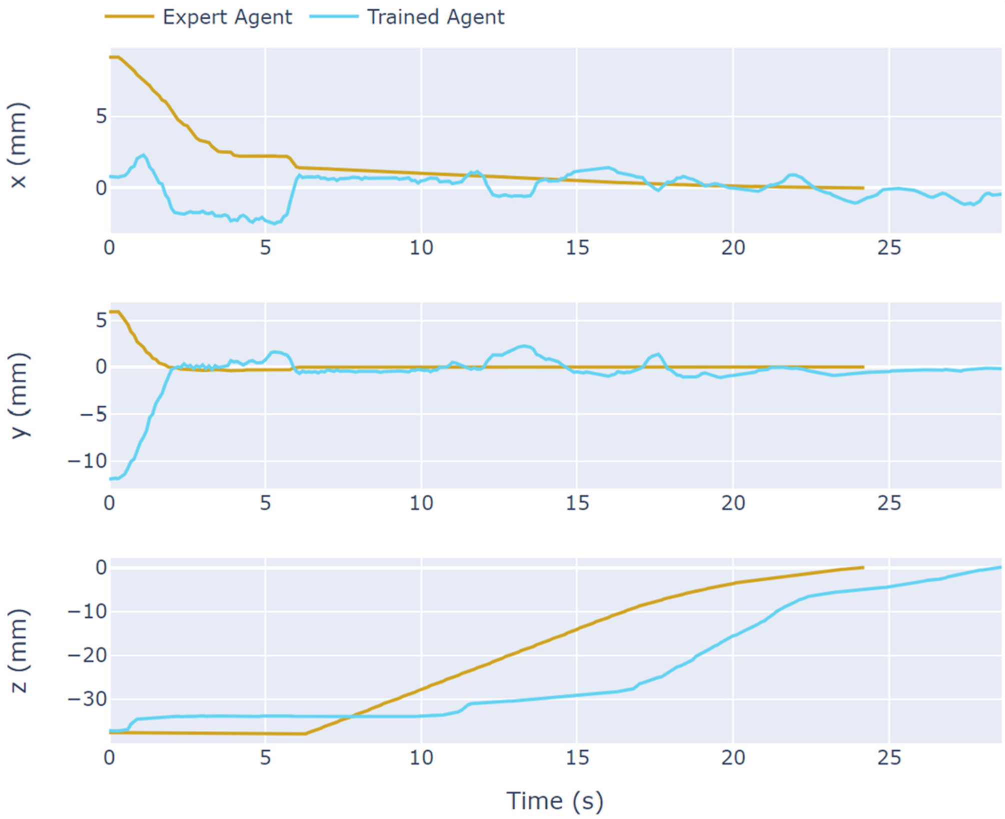


Figure 11. Expert and trained agent (*vq-rec*) trajectories on rollouts with the same environment conditions. Trained agent predictions are produced online by rolling out a new episode.

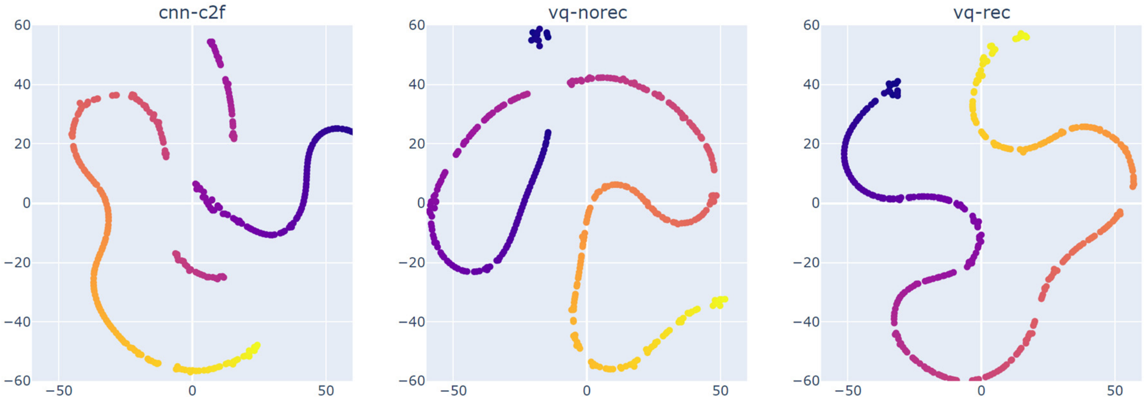


Figure 12. t-SNE mapping of embeddings produced by the three IL approaches considered. Color shade indicates the episode step index.

As seen in Figure 12, there are significant differences regarding the smoothness of the embedding manifold; the embeddings produced by the model without Vector Quantization or image reconstruction are significantly fragmented, i.e., there are exceptional gaps leading to observations that are close to each other temporally to be mapped far away from each other. The introduction of Vector Quantization significantly improves this mapping, reducing the gaps between episode steps while image reconstruction further enhances this aspect. We theorize that this smoothing effect is the main reason behind the improvement in the model performance.

It is worth noting that even though the actuation of the gripper is not learned by the proposed approach, as the manipulation sequence is just a replication of the expert agent demonstration, accurately positioning the gripper presents challenges that are not normally observed in rigid grippers. An example is illustrated in Figure 13, where two screenshots belonging to two different episode rollouts are shown. The rectangular annotations are placed at the exact same pixel coordinates in both images to highlight that, in contrast to a rigid gripper, whose fingers are expected to show at the exact same location during visual servoing, this is not the case for the soft fingers. The latter are prone to displacements due to hysteresis or because of collisions with other objects. Given the fact that our approach is learning the visual servoing function in an end-to-end fashion, i.e., directly from pixels, such discrepancies present an additional challenge.

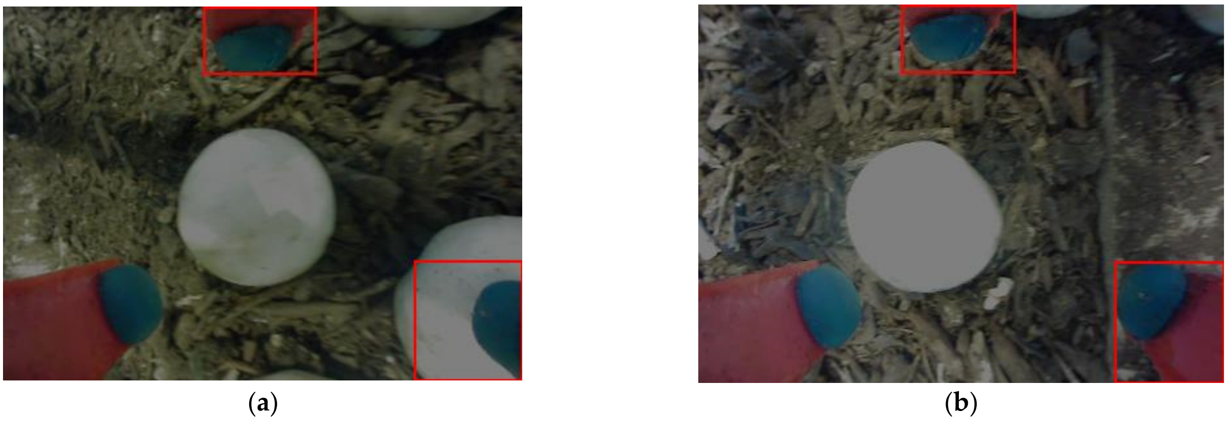


Figure 13. Annotated screenshots during trained agent (*vq-rec*) episode rollouts: (**a**) the bottom right finger is significantly displaced due to collision with a nearby mushroom, (**b**) the top finger is moderately displaced due to hysteresis. The rectangular red annotations are placed at exactly the same pixel coordinates to highlight these discrepancies.

5. Discussion

We have presented a pixel-based, one-shot IL approach, leveraging Vector Quantization, for learning to solve the mushroom harvesting task. The approach was tested on a lab-scale mushroom picking environment involving a cartesian robot and a soft, pneumatically actuated gripper. The trials were designed based on observations from real mushroom harvesting experiments. We demonstrated the benefit of using Vector Quantization and showed that the proposed model architecture can learn the desired behavior with minimal input, namely a simple RGB camera and the z-position encoder measurement of the robot. Our approach operates directly on raw observations with no pre-processing other than downscaling. By casting the problem as a primarily visual servoing task, our approach can learn the controller for reaching the optimal position for grasping the mushroom based on less than 20 min of data collection. The approach is shown to solve the task in 100% of the cases, even though mushrooms are picked from a cluster of very similar distractors. It is also computationally lean, with the entire pipeline consisting of <800 k parameters, allowing for inference of ~20 ms on a moderate laptop GPU (GTX 1650 Ti). The low computational requirements coupled with the low cost of the sensorial streams required for our pipeline make for a practical method and system for IL in an industrial setting.

A crucial aspect of our approach is that it gracefully handles the lack of proprioception information about the finger position. It is also shown to be exceptionally robust to the challenges presented by the soft fingers' displacements due to hysteresis or collisions, paving the way for a straightforward learning of harvesting tasks with soft grippers.

In the future, we plan to evaluate the effectiveness of our approach on large-scale mushroom growing facilities. Within this context, we aim to integrate further inputs into the system such as tactile sensing. A significant step forward would also be to consider cases where the mushrooms are more tightly clustered together or the cap orientation significantly deviates from the common case of vertical pose. Speeding up the visual servoing process and deploying multiple grippers will also be of paramount importance to match human picking performance.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/robotics13070098/s1, Video S1: Imitation Learning from a Single Demonstration Leveraging Vector Quantization for Robotic Harvesting.

Author Contributions: Conceptualization, A.P.; data curation, N.K.; formal analysis, A.P.; investigation, A.P., M.I. and N.K.; methodology, A.P.; writing—original draft preparation, A.P.; writing—review and editing, V.M. and P.C.; visualization, A.P.; supervision, V.M.; project administration, P.C.; funding acquisition, A.P. and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the EU's Horizon 2020 research and innovation program, under grant agreement No. 101017054 (Softgrip).

Data Availability Statement: Data are available upon request.

Acknowledgments: This publication was made possible by the sponsorship and support of Lloyd's Register Foundation. Lloyd's Register Foundation helps to protect life and property by supporting engineering-related education, public engagement, and the application of research. The work was enabled through, and undertaken at, the National Structural Integrity Research Centre (NSIRC), a postgraduate engineering facility for industry-led research into structural integrity established and managed by TWI through a network of both national and international universities. We would also like to thank Andrea Uccello and Helen Grogan from Teagasc (Oak Park, Carlow, R93 XE12, IE) for their invaluable support in terms of fresh mushroom harvesting demonstrations and information on related challenges.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Duan, Y.; Chen, X.; Edu, C.X.B.; Schulman, J.; Abbeel, P.; Edu, P.B. Benchmarking Deep Reinforcement Learning for Continuous Control. *arXiv* **2016**, arXiv:1604.06778.
2. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent Advances in Robot Learning from Demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [CrossRef]
3. Huang, M.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Picking dynamic analysis for robotic harvesting of Agaricus bisporus mushrooms. *Comput. Electron. Agric.* **2021**, *185*, 106145. [CrossRef]
4. Carrasco, J.; Zied, D.C.; Pardo, J.E.; Preston, G.M.; Pardo-Giménez, A. Supplementation in Mushroom Crops and Its Impact on Yield and Quality. *AMB Express* **2018**, *8*, 146. [CrossRef] [PubMed]
5. Yang, S.; Ji, J.; Cai, H.; Chen, H. Modeling and Force Analysis of a Harvesting Robot for Button Mushrooms. *IEEE Access* **2022**, *10*, 78519–78526. [CrossRef]
6. Mohanan, M.G.M.M.G.; Salgaonkar, A.S.A. Robotic Mushroom Harvesting by Employing Probabilistic Road Map and Inverse Kinematics. *BOHR Int. J. Internet Things Artif. Intell. Mach. Learn.* **2022**, *1*, 1–10. [CrossRef]
7. Yin, H.; Yi, W.; Hu, D. Computer Vision and Machine Learning Applied in the Mushroom Industry: A Critical Review. *Comput. Electron. Agric.* **2022**, *198*, 107015. [CrossRef]
8. Mavridis, P.; Mavrikis, N.; Mastrogeorgiou, A.; Chatzakos, P. Low-Cost, Accurate Robotic Harvesting System for Existing Mushroom Farms. *IEEE/ASME Int. Conf. Adv. Intell. Mechatron. AIM* **2023**, *2023*, 144–149. [CrossRef]
9. Bissadu, K.D.; Sonko, S.; Hossain, G. Society 5.0 Enabled Agriculture: Drivers, Enabling Technologies, Architectures, Opportunities, and Challenges. *Inf. Process. Agric.* 2024; *in press*. [CrossRef]
10. Johns, E. Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration. *Proc. IEEE Int. Conf. Robot. Autom.* **2021**, *2021*, 4613–4619. [CrossRef]
11. Porichis, A.; Vasios, K.; Iglezou, M.; Mohan, V.; Chatzakos, P. Visual Imitation Learning for Robotic Fresh Mushroom Harvesting. In Proceedings of the 2023 31st Mediterranean Conference on Control and Automation, MED 2023, Limassol, Cyprus, 26–29 June 2023; pp. 535–540. [CrossRef]
12. Ng, A.Y.; Russel, S.J. Algorithms for Inverse Reinforcement Learning. In Proceedings of the ICML '00 17th International Conference on Machine Learning, San Francisco, CA, USA, 29 June–2 July 2000; pp. 663–670.
13. Finn, C.; Levine, S.; Abbeel, P. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. *arXiv* **2016**, arXiv:1603.00448.
14. Das, N.; Bechtle, S.; Davchev, T.; Jayaraman, D.; Rai, A.; Meier, F. Model-Based Inverse Reinforcement Learning from Visual Demonstrations. *arXiv* **2021**, arXiv:2010.09034.
15. Haldar, S.; Mathur, V.; Yarats, D.; Pinto, L. Watch and Match: Supercharging Imitation with Regularized Optimal Transport. *arXiv* **2022**, arXiv:2206.15469. [CrossRef]
16. Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; Levine, S.; Brain, G. Time-Contrastive Networks: Self-Supervised Learning from Video. In Proceedings of the IEEE International Conference on Robotics and Automation, Honolulu, HI, USA, 21–26 July 2017; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018; pp. 1134–1141.

17. Ho, J.; Ermon, S. Generative Adversarial Imitation Learning. *Adv. Neural. Inf. Process Syst.* **2016**, *29*, 4572–4580.
18. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2014**, *63*, 139–144. [CrossRef]
19. Dadashi, R.; Hussenot, L.; Geist, M.; Pietquin, O.; Wasserstein, O.P.P. Primal Wasserstein Imitation Learning. In Proceedings of the ICLR 2021—Ninth International Conference on Learning Representations, Virtual, 4 May 2021.
20. Pomerleau, D.A. ALVINN: An Autonomous Land Vehicle in a Neural Network. *Adv. Neural. Inf. Process Syst.* **1988**, *1*, 305–313.
21. Rahmatizadeh, R.; Abolghasemi, P.; Boloni, L.; Levine, S. Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-to-End Learning from Demonstration. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3758–3765. [CrossRef]
22. Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O.A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; Tompson, J. Implicit Behavioral Cloning. In Proceedings of the 5th Conference on Robot Learning, PMLR, Zurich, Switzerland, 29–31 October 2018; pp. 158–168.
23. Janner, M.; Li, Q.; Levine, S. Offline Reinforcement Learning as One Big Sequence Modeling Problem. *Adv. Neural. Inf. Process Syst.* **2021**, *2*, 1273–1286. [CrossRef]
24. Shafiullah, N.M.; Cui, Z.; Altanzaya, A.A.; Pinto, L. Behavior Transformers: Cloning $ k $ Modes with One Stone. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 16 May 2022; pp. 22955–22968.
25. Zhao, T.Z.; Kumar, V.; Levine, S.; Finn, C. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In Proceedings of the Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea, 23 April 2023.
26. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In Proceedings of the 6th Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
27. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. *arXiv* **2020**, arXiv:2006.11239.
28. Pearce, T.; Rashid, T.; Kanervisto, A.; Bignell, D.; Sun, M.; Georgescu, R.; Macua, S.V.; Tan, S.Z.; Momennejad, I.; Hofmann, K.; et al. Imitating Human Behaviour with Diffusion Models. *arXiv* **2023**, arXiv:2301.10677.
29. Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *arXiv* **2023**, arXiv:2303.04137. [CrossRef]
30. Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 3942–3951. [CrossRef]
31. Vitiello, P.; Dreczkowski, K.; Johns, E. One-Shot Imitation Learning: A Pose Estimation Perspective. *arXiv* **2023**, arXiv:2310.12077.
32. Valassakis, E.; Papagiannis, G.; Di Palo, N.; Johns, E. Demonstrate Once, Imitate Immediately (DOME): Learning Visual Servoing for One-Shot Imitation Learning. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Kyoto, Japan, 23–27 October 2022; pp. 8614–8621. [CrossRef]
33. Park, J.; Seo, Y.; Liu, C.; Zhao, L.; Qin, T.; Shin, J.; Liu, T.-Y. Object-Aware Regularization for Addressing Causal Confusion in Imitation Learning. *Adv. Neural. Inf. Process Syst.* **2021**, *34*, 3029–3042.
34. Kujanpää, K.; Pajarinen, J.; Ilin, A. Hierarchical Imitation Learning with Vector Quantized Models. In Proceedings of the International Conference on Machine Learning, Chongqing, China, 27–29 October 2023.
35. Van Den Oord, A.; Vinyals, O.; Kavukcuoglu, K. Neural Discrete Representation Learning. *arXiv* **2017**, arXiv:1711.00937. [CrossRef]
36. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural. Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
37. Chen, X.; Toyer, S.; Wild, C.; Emmons, S.; Fischer, I.; Research, G.; Lee, K.-H.; Alex, N.; Wang, S.; Luo, P.; et al. An Empirical Investigation of Representation Learning for Imitation. *arXiv* **2021**, arXiv:2205.07886.
38. Pagliarani, N.; Picardi, G.; Pathan, R.; Uccello, A.; Grogan, H.; Cianchetti, M. Towards a Bioinspired Soft Robotic Gripper for Gentle Manipulation of Mushrooms. In Proceedings of the 2023 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor, Pisa, Italy, 6–8 November 2023; pp. 170–175. [CrossRef]
39. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. *Ultralytics/Yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*; Zenodo; CERN: Geneva, Switzerland, 2022. [CrossRef]
40. van der Maaten, L.; Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

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.