POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Autonomous robotic strawberry harvesting: from perception to trajectory planning

Laurea Magistrale in Mechanical Engineering - Ingegneria Meccanica

Author: Alessandra Tafuro

Advisor: Prof. Andrea Maria Zanchettin

Co-advisor: Prof. Amir M. Ghalamzan

Academic year: 2020-2021

1. Introduction

The worldwide demand for agricultural products is rapidly increasing due to the growing population, but, in parallel, rural human labor shortage due to different factors is becoming a limit. Moreover, the recent Covid-19 pandemic has shown how possible travel restrictions can limit the affluence of seasonal farmworkers. This means that the agricultural chain is still strongly dependent on human labor, which is very risky in the current era. The automation of agriculture can be the solution to tackle the increasing load on farming businesses. Despite several attempts to develop a robotic solution for harvesting strawberries and other crops, a fully viable commercial robotic system has yet to be established. This thesis deals with some main problems for the development of a strawberry harvesting robotic technology: ready-to-be-picked strawberries detection, fruit weight estimation before picking, and path planning from visual information to reach the target fruit.

2. Related Works

Fruit detection: Different machine vision systems for fruit localisation exist since it is a fundamental part to be developed for agriculture robotization. For example, some famous works exploited the conversion of strawberries images from RGB to HSI colour map to manually set a threshold to identify ripe berries. However, the inability to generalise and being prone to noise are among the weaknesses of colour thresholding, geometry-based algorithms, and other traditional approaches. Thus, authors begun to adopt some Deep Learning (DL) techniques for fruit perception. Some researchers utilized Convolutional Neural Networks (CNNs) to calculate the relative 3-D location of fruit. CNNs perform well in image-specific tasks such as classification, but for the pixel-wise understanding of images (semantic segmentation) Regional-CNNs (RCNNs) are preferred. Mask-RCNN (MRCNN) has been implemented in some public works to determine strawberries' shapes and to localise the picking point. MRCNN is the de facto standard for successful object identification, and this is the reason why it has been chosen in this work for strawberry perception and key-points detection.

Fruit weight estimation: Another important need for robotic automation in agriculture is a system able to determine the volume, area, and mass information of agricultural products, rely-

ing on visual data only. Both classic computer vision (CV) techniques and DL have been exploited to this end (for example determining the linear relation between the measured area and the actual weight of the mangoes, or regressing the food volume trough CNNs). Based on these works an original strawberry weight estimation method has been developed, exploiting both state-of-the-art machine learning (ML) and DL techniques.

Path planning: Learning from demonstration (LfD) is a method used for training the robot to perform a certain task (in this case it would be to reach a certain ripe target berry to be picked) with several demonstrations performed by an expert. Dynamic Movement Primitives (DMP) is a well-known LfD approach able to encode the desired motion to be learned with a certain set of parameters (weights). Recently, DL has been used to generate the DMP parameters directly from an image of the environment in which the movement has to be executed. If there is some variability in the execution of a certain task (as in the strawberry picking case, where the fruit can be approached in multiple ways), it can be captured with the Probabilistic Movement Primitives (ProMPs) framework, able to represent the probability distribution of a set of demonstrations with a distribution of weights. The path planning approach that is proposed, called ***Deep Probabilistic Movement Primitives (Deep-ProMP)***, maps the visual information into a distribution of robot trajectories expressed by the ProMPs weights. Deep-ProMP has a two-fold design: from the input image to a latent representation and from the latent representation to the desired trajectory. Moreover it has been designed exploiting the architecture of Autoencoder (AE), Variational Autoencoder (VAE)or conditional Variational Autoencoder (cVAE). Finally, being inspired by [2] regularisation and domain-specific training have been implemented to improve the latent space representation of the input image.

3. Fruit detection trough key-points

3.1. Datasets and proposed approach

To determine picking points and suitability for picking, a DL-based key-points detection approach, which has been successfully applied in other domains, e.g. face landmark detection, has been utilized. The proposed approach includes ***Detectron-2*** [5], an open-source object detection system from Facebook AI Research. It is based on MRCNN and has become the de facto standard for instance segmentation. Experiments with three backbone networks, R50-FPN, X101, and X101-FPN have been performed. Two novel datasets have been created to train the model since the public strawberries datasets do not include key-points annotations. ***Dataset-1*** is collected at a new 15-acre strawberry glasshouse in Carrington, Lincolnshire, and presents strawberries' weights, suitability for picking, instance segmentation, and key-points for grasping and picking action. ***Dataset-2*** has been derived from the public Strawberry Digital Images (SDI) [4] dataset adding the key-points annotations. For each strawberry, the datasets present annotations for five different key-points: picking point (PP), top and bottom points, left and right grasping points (LGP, RGP) (Fig. 1). Each strawberry is labeled as *"pluckable"*–ready to be picked– or *"unpluckable"*–not to be picked–. In total, Dataset-1 and Dataset-2 include 1588 and 3100 strawberries images respectively. Table 1 summarises the results for segmentation and key-points detection for both the datasets with Detectron-2 [5]. X101-FPN and X101-based models perform better than R50-FPN based model. The first two columns show segmentation Average Precision (AP) values for "pluckable" and "unpluckable" berries. The sub-columns show AP for Intersection over Union (IoU) thresholds of 0.5, 0.7, and 0.9. Using Dataset-2 decent AP values are obtained at IoU 0.5 but the performance drops significantly for stricter IoU 0.7 and 0.9. Dataset-1 shows very reliable AP values for "pluckable" strawberries across IoU thresholds. With IoU threshold of 0.5, 93.32 (R50-FPN), and (X101-FPN) 94.19 AP values are ob-

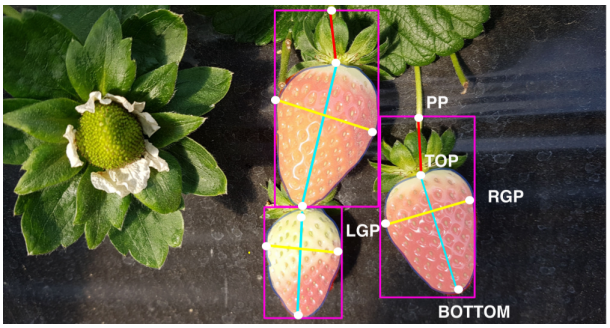

Figure 1: Strawberry key-points.

Table 1: Segmentation and key-points detection results with Detectron-2.

Dataset	Backbone	Segm "pluck"			Segm "unpluck"			KP "pluck"			KP "unpluck"		
		0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
1	R50-FPN	93.3	90.9	83.5	59.4	53.6	42.9	91.2	89.1	81.9	51.3	46.2	37.3
	X101	94.1	92.8	88.7	61.1	56.2	45.6	92.7	91.4	87.7	61.2	56.5	46.8
2	X101-FPN	71.1	64.7	43.2	76.8	74.5	68.7	64.3	58.9	39.9	73.2	71.3	66.4
	X101	72.1	66.8	47.8	78.0	76.6	70.3	59.2	54.4	42.1	74.6	71.4	65.3

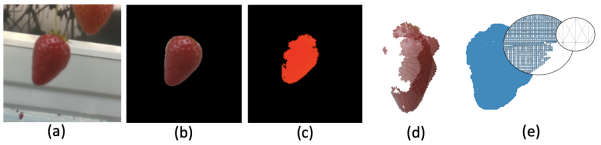

Figure 2: Strawberry RGB (a), segmented RGB (b) and depth (c) images, point cloud (d) and graph (e).

tained, while with IoU of 0.9, Detectron-2 provides AP of 83.55 (R50-FPN) and 88.70 (X101-FPN). This shows that for selective harvesting the proposed datasets can be reliably used. For Dataset-1, the performance on "unpluckable" berries is comparatively less reliable as there are fewer samples of "unpluckable" berries in this dataset, while the situation is reversed in Dataset-2. The results of the key-points detection expressed in terms of AP at different OKS thresholds (0.5, 0.3, and 0.1) are similar to segmentation. OKS and IoU are the standard performance metrics used by MSCOCO [3] for key-point detection and segmentation. The experimental results are consistent across the two backbones although X101-FPN performs slightly better.

4. Berries weight estimation

4.1. Implemented solutions

For strawberry weight estimation, different state-of-the-art neural networks have been trained. Dataset-1 has been used for this purpose since it contains the annotations of the berries' weights. First, all the strawberries instances from each RGB image have been extracted through Detectron-2 (Fig. 2.b). The segmentation mask is also applied to the depth image (Fig. 2.c). These two segmented images (color and depth) are then combined to reconstruct the point cloud (Fig. 2.d). This is fed into PointConv , PointNet and PointNet++, which are well-known point cloud-based deep networks. Recently there has been an increased interest in graph-based neural networks, thus DGCN , GCN and HGNN have also been tested. The graph dataset (Fig. 2.e) is obtained starting from the point clouds exploiting the k-nearest neighbor graph generation function.

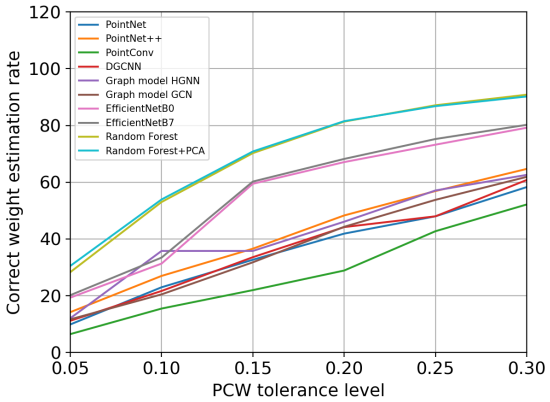

Figure 3: Results of weight estimation.

Also, the well-known EfficientNet has been implemented using a two-stream architecture fed with RGB and depth data. However, the model which turned out to perform better is the simpler ***Random Forest model*** [1] ***with Decision Trees***. It is fed with a feature vector with the strawberry bounding box area, the segmentation mask, the histogram of depth values, and the point cloud primary principal components (PCA). The inclusion of the PCA improves the weight estimation since it gives the model insights into the 3-D orientation of the berry, which causes the variation of the apparent size of the fruit segmentation mask. To measure the accuracy of the different models, the Percentage of Correct Weights (PCW@tol) protocol has been proposed, which measures the regression error in percentage to the ground truth. The percentage of predictions within the tolerance (tol) values gives the model accuracy. Figure 3 illustrates the result of strawberry weight estimation experiments. PointConv provides only 15% (PCW @0.1) to 28% (PCW @0.2). PointNet and PointNet++ perform in similar way, as well as graph-based networks. EfficientNet B0 and B7 give 67% and 68% accuracy at PCW @0.2, still far from suitability for selective harvesting but better than point cloud and graph-based networks. This motivates the proposal of the most accurate Random Forest model with Decision Trees (51% at PCW @0.1 and 23% at PCW @0.2). The main novelty is that the inclusion of PCA helps with slightly more accurate weight estimation (e.g. the performance improves from 28%

to 29% at PCW @0.05).

5. Path planning trough LfD

After the recognition of the target fruit, the robotic system ultimate goal is the ability to efficiently reach-to-pick harvest-ready strawberries. This is solved in a LfD setting and exploiting the ProMP formulation.

5.1. Problem formulation

The problem is formulated in joint space, but it can be easily extended to task space. Let's consider a set of N_{tr} demonstrations $\mathcal{T} := \{\{\mathbf{Q}^1, \mathbf{I^1}\}, \ldots, \{\mathbf{Q}^{N_{\text{tr}}}, \mathbf{I^{N_{\text{tr}}}}\}\}$ for the reach-to-pick task. $\mathbf{Q}^n$ are the joints sets of trajectories, and $\mathbf{I^n}$ is the RGB image of the robot's workspace. A set of trajectories instead of a single one is collected since the probabilistic face of the behaviour should be captured. A set of joint trajectories is defined as per Eq. 1.

$$\mathbf{q_j} := \left\{\mathbf{q}_s^j\right\}_{s=1,\ldots,S} := \left\{q_{t,s}^j\right\}_{t=1,\ldots,T;s=1,\ldots,S} \tag{1}$$

$q_{t,s}^j \in R$ is the joint position during trial s at time instant t. Considering all the joints together: $\mathbf{Q} := \{\mathbf{q}_1, ..., \mathbf{q}_{N_{\text{joint}}}\}$. The ProMPs framework is exploited to represent the demonstrated sets of trajectories. The robot single trajectory is described as per Eq. 2, where ψ_i are basis functions (Gaussian for stroke-like movements) evaluated at $z(t)$, z is a phase function that allows time modulation, $\theta_i \in R$ are the weights and an observation uncertainty $\epsilon_{\mathbf{q}_s^j}$ adds zero-mean Gaussian observation noise with variance $\Sigma_{\mathbf{q}_s^j}$.

$$\mathbf{q}_s^j = \sum_{i=1}^{N_{\text{bas}}} \theta_i \psi_i(z(t)) + \epsilon_{\mathbf{q}_s^j} \tag{2}$$

Eq. 2 can be written in matrix form $\mathbf{q}_{t,s}^j = \mathbf{\Psi}_t^T \mathbf{\Theta_j^s} + \epsilon_{q_{t,s}^j}$ where $\mathbf{\Psi}_t := (\psi_1(z(t)), \ldots, \psi_{N_{\text{bas}}}(z(t))) \in R^{N_{\text{bas}} \times 1}$, $\mathbf{\Theta_j^s} := (\theta_1, \ldots, \theta_{N_{\text{bas}}}) \in R^{N_{\text{bas}} \times 1}$, $\mathbf{\Omega} := (\mathbf{\Theta^s}_1, \ldots, \mathbf{\Theta^s}_{N_{\text{joint}}}) \in R^{N_{bas} N_{\text{joint}} \times 1}$ and $\mathbf{\Phi} := [\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_T]^T \in R^{T \times N_{\text{bas}}}$. It follows from Eq. 2 that the probability of observing $q_{t,s}^j$ is given by $p(q_{t,s}^j|\mathbf{\Theta}) = \mathcal{N}\left(q_{t,s}^j \,\middle|\, \mathbf{\Psi}_t^T \mathbf{\Theta_j^s}, \Sigma_{q_{t,s}^j}\right)$. $\Sigma_{q_{t,s}^j}$ is the same for every time step t and every trial s ($\Sigma_{q_{t,s}^j} = \Sigma_{q_j}$) so the values $q_{t,s}^j$ are taken from independent and identical distributions. It can be assumed that the weight parameters are taken from a distribution, thus, the distribution of $q_{t,s}^j$, which does not depend on $\mathbf{\Theta_j^s}$, but on $\rho := (\mathbf{\Theta}_{mean,j}, \mathbf{\Sigma}_{\Theta_j})$, can be estimated.
This means that the demonstrated trajectory distribution for joint j can be represented by its mean and covariance values $(\mathbf{q}_{mean,j}, \mathbf{\Sigma}_{q_j})$, which in turn can be derived by the mean and covariance values of the ProMPs weights $(\mathbf{\Theta}_{mean,j}, \mathbf{\Sigma}_{\Theta_j})$, as described in Eq. 3.

$$\begin{aligned} \mathbf{q}_{mean,j} &= \mathbf{\Phi}^T \mathbf{\Theta}_{mean,j}, \\ \mathbf{\Sigma}_{q_j} &= \mathbf{\Phi}^T \mathbf{\Sigma}_{\Theta_j} \mathbf{\Phi} \end{aligned} \tag{3}$$

Deep-ProMP is the proposed probabilistic deep model that maps visual information to the distribution of robot trajectories. For each joint j the relative trajectory distribution can be expressed in weights space with $\boldsymbol{\Theta_{mean,j}}$ and $\boldsymbol{\Sigma_{\Theta,j}}$. The deep model learns the relation between these two parameters and the input image.

$$\boldsymbol{\Theta_{mean,j}}, \mathbf{\Sigma}_{\boldsymbol{\Theta_j}} = \mathrm{f}_j(\hat{\boldsymbol{W}}_{\boldsymbol{j}}, \mathbf{I}^n, \hat{\sigma}_j) \tag{4}$$

Eq. (4) shows that $\boldsymbol{\Theta_{mean,j}}$, $\mathbf{\Sigma}_{\boldsymbol{\Theta,j}}$ are equivalent to a non-linear deep model ($\mathrm{f_j}$) of the input image $\mathrm{I^n}$, the weight parameter $\hat{\boldsymbol{W}}_{\boldsymbol{j}}$ and the node activation $\hat{\sigma}_\mathrm{j}$. The predicted weights distribution generates the corresponding trajectories distribution using Eq. (3). Different baselines for Deep-ProMP models architectures have been proposed to improve the accuracy in the prediction of the demonstrated behaviour, but all have two parts: (1) **part-1** encodes the high-dimensional input RGB image in a low-dimensional latent space vector, preserving all the relevant information, using a set of convolutional layers, as per Eq. (5);

$$\mathbf{E^n} = Encoder(\mathbf{W}_{enc}, \mathrm{I^n}, \sigma_{enc}) \tag{5}$$

(2) **part-2** maps the latent embedded vector to the relative ProMP weights distribution using a Multi Layer Perceptron (MLP) (one for each joint) as per Eq. (6).

$$\hat{\mathbf{\Theta}}_j, \hat{\mathbf{\Sigma}}_{\Theta_j} = h_j(\mathbf{W}_j, \mathbf{E^n}, \sigma_j) \tag{6}$$

This twofold design of the model has been proven to have an advantage over a direct mapping of the image to the trajectories distributions. The first deep-ProMP architecture (**deep-ProMP-AE**) uses the encoding layers (*Encoder*) of an AE network for part-1 to reduce the input dimensionality while preserving the important information. In the second baseline (**deep-ProMP-VAE**), the latent representation $\mathbf{E^n}$ is stochastic ($\mathbf{E^n} \sim \mathcal{N}(\mu_{\boldsymbol{E^n}}, \Sigma_{\boldsymbol{E^n}})$)

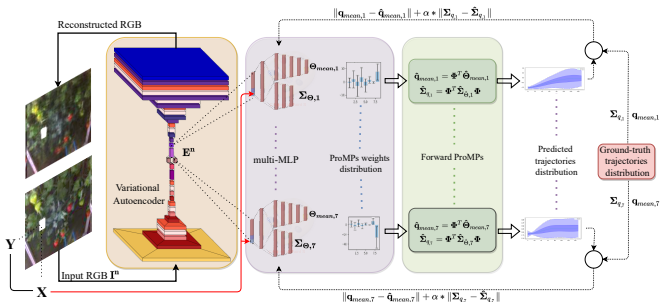

Figure 4: Deep-ProMP-cVAE (tr-1).

and obtained training a VAE. Part-2 of both models is a MLP that maps the deterministic or stochastic latent representation to the ProMP weights distributions as per Eq. (6). In the third model (**deep-ProMP-cVAE**), a conditional variable **c** is concatenated with the latent vector to help the consequent MLP to tailor its behaviour according to some abstract information, e.g., the pixel coordinate of a target berry (Fig. 4). There are two ordered training stages for the previous models: **(tr-1)** unsupervised training of part-1 (using image reconstruction loss to train AE or VAE) and **(tr-2)** supervised training of part-2 to train MLPs (one for each joint) using the loss in Eq. 7, where α is a tuning parameter that weights the loss components.

$$e = \|\mathbf{q}_{mean,j} - \hat{\mathbf{q}}_{mean,j}\| + \alpha\|\mathbf{\Sigma}_{q,j} - \hat{\mathbf{\Sigma}}_{q,j}\| \quad (7)$$

tr-1 and tr-2 are completely decoupled, so the latent space maintains the information necessary for RGB image reconstruction independently from the trajectories distributions. While this non-domain-specific training is useful for CV, it is not relevant for robotic tasks. Hence, it has been proposed to continue the training of the weights of *Encoder* using the loss in Eq. 8 while the MLP part is kept fixed. This is called **domain-specific latent space training**. In this way, there is a direct mapping of the latent space to the information useful both for image reconstruction and trajectory prediction.

5.2. Experimental results

To validate the models, the experimental setup consists of a 7-DoF Panda robotic arm manufactured by Franka Emika with a custom gripper specific for strawberry picking. An Intel RealSense D435i RGB-D camera is mounted on the top of the gripper. A mock set up with plastic strawberries is used. 250 demonstrations samples have been collected where each sample includes an RGB image (VGA resolution) of the scene and the robot trajectory starting

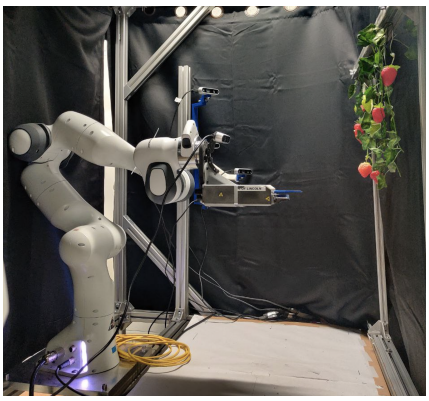

Figure 5: Robotic arm in home position.

from a home configuration (Fig. 5). After taking an image, the robot is manually moved to reach the target berry in kinesthetic teaching mode. The target berry in the input RGB image is masked with a white bounding box. The movement is repeated 10 times for a single target strawberry to capture the demonstrations' variations. 5 different strawberry plant configurations, each including 5 target berries have been created. Deep-ProMP-AE, Deep-ProMP-VAE, and Deep-ProMP-cVAE (which is conditioned concatenating the pixel coordinate of the target berry bounding box center to the VAE latent vector) have been trained to make predictions both in task and joint space. Moreover, the domain-specific latent space learning of these three models (l-Deep-ProMP-AE, l-Deep-ProMP-VAE, l-Deep-ProMP-cVAE) has been implemented using the prediction error loss as per Eq. (8).

$$\begin{aligned} e = \|\mathbf{q}_{mean,j}(t_{end}) - \hat{\mathbf{q}}_{mean,j}(t_{end})\| + \\ \alpha\|\mathbf{\Sigma}_{q,j}(t_{end}) - \hat{\mathbf{\Sigma}}_{q,j}(t_{end})\| \end{aligned} \quad (8)$$

For the first experiment, the prediction performance of the 7 joint trajectories distributions has been evaluated on a test set never seen in training or validation stages. The evaluation metric is the loss in Eq. 7. It has been noticed that the prediction error improves from Deep-ProMP-AE, to Deep-ProMP-VAE (-62% of Deep-ProMP-AE error), to Deep-ProMP-cVAE (-70% of Deep-ProMP-AE error). Hence, *Deep-ProMP-cVAE is outperforming all others.* The same observation applies in task space (Table. 2). Another experimentation on the same test set has been done to compare the model performances between task and joint space predictions. The evaluation metric in Eq. 9 has been used.

$$\begin{aligned} e_{pos} = \sqrt{(x-\hat{x})^2 + (y-\hat{y})^2 + (z-\hat{z})^2} \\ e_{ori} = min[\|q-\hat{q}\|, \|q+\hat{q}\|] \end{aligned} \quad (9)$$

Table 2: Task space predictions.

Task	dP-AE	dP-VAE	drop	dP-cVAE	drop
X	1.4×10^{-4}	$\mathbf{1.4\times10^{-4}}$	-0%	$\mathbf{1.4\times10^{-4}}$	-0%
Y	15.3×10^{-4}	7.0×10^{-4}	-53%	$\mathbf{0.8\times10^{-4}}$	-88%
Z	0.9×10^{-4}	$\mathbf{0.1\times10^{-4}}$	-88%	$\mathbf{0.1\times10^{-4}}$	-0%
Q1	1.5×10^{-4}	$\mathbf{0.5\times10^{-4}}$	-68%	$\mathbf{0.5\times10^{-4}}$	-0%
Q2	$24.\times10^{-4}$	4.1×10^{-4}	-83%	$\mathbf{1.5\times10^{-4}}$	-75%
Q3	1.1×10^{-4}	0.3×10^{-4}	-80%	$\mathbf{0.2\times10^{-4}}$	-43%
Q4	0.7×10^{-4}	0.3×10^{-4}	-68%	$\mathbf{0.3\times10^{-4}}$	-8%

In Eq. 9, (x, y, z) and $(\hat{x}, \hat{y}, \hat{z})$, and q and $\hat{q}$ represent the ground truth and predicted positions and orientation (quaternions) of the end effector at the final time step. *Task space predictions are 50% more accurate than joint space ones.* Moreover, the most accurate model is again Deep-ProMP-cVAE. The third set of experiments has been done using the real robot. 5 new (different) strawberry plant configurations each including again 5 different target berries have been created. This demonstrates the generalisation ability of the model in predicting the reaching movement in unseen settings. The robot has been firstly moved to the desired final pose necessary for picking a target berry and the (x, y, z) position has been recorded, to have a reference for evaluation. The predicted mean trajectory together with the trajectory at 2σ and at -2σ from the mean have been evaluated. e_{pos} in Eq. (9) has been used as metric. The predictions have been made in task space since they have been proven to be more accurate. Fig. 6 shows that *the model performance increases after the domain-specific training.* Furthermore, the most accurate model is l-Deep-ProMP-cVAE. The mean predicted trajectory performance is always better than the trajectories sampled at some σ from the mean. Additionally, the probabilistic framework has been exploited to perform the task in different ways; for example, the robot can reach the target point with different orientations sampling from the predicted quaternion distribution. Finally, the clustering level of the latent space before and after the domain-specific training has been investigated. Table 3 shows the Davies-Bouldin score which indicates the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, the lower the scores, the higher the level of clustering of the latent space. *The clustering level increases from AE to VAE and it increases even more with the domain-specific latent space learning.*

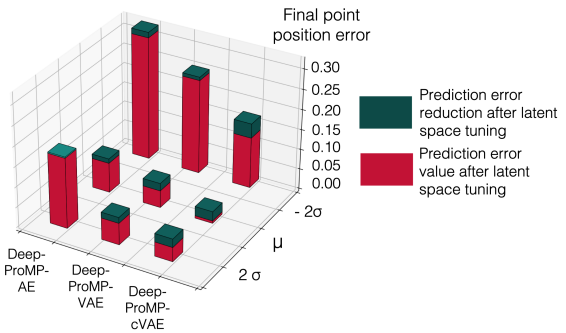

Figure 6: Real robot experimental results.

Table 3: Davies-Bouldin Index.

		DB score
AE	base	0.616
	latent space tuned	**0.482**
VAE	base	0.424
	latent space tuned	**0.352**

6. Conclusions

The increase in the worldwide demand for agricultural products due to the growing population coupled with labor shortage is an issue that automation in the field of agriculture can solve. This thesis deals with some main problems for a successful robotic strawberries harvesting technology: ready-to-be-picked strawberry segmentation, strawberry weight estimation, and path planning from visual information to reach the target fruit. The first problem has been addressed with Detectron-2 [5]. In particular, it has been trained to segment berries, classify them as ripe or unripe, and detect the key-points necessary for picking and grasping action. Moreover, two new datasets useful for selective harvesting of strawberries have been presented. Strawberry weight estimation is achieved training a Random Forest Model [1] with Decision Trees. This approach outperforms many state-of-the-art methods. Finally, a novel framework named deep Probabilistic Movement Primitives which maps the visual information of a robot's workspace into the corresponding robot trajectories, according to a set of human expert demonstrations has been presented. A few model architectures have been proposed (Deep-ProMP-AE, Deep-ProMP-VAE, and Deep-ProMP-cVAE). In addition, a novel domain-specific latent space training has been implemented. This allows learning a latent space that is relevant both for the specific robotic task and CV. The results suggest that the deep-ProMP conditioning with a relevant feature and domain-specific training of the latent space yields the best performances.

References

[1] L. Breiman. Random forests. In *Machine Learning 45(1), 5-32*, 2001.

[2] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2015.

[4] Isaac Pérez-Borrero, Diego Marín-Santos, Manuel E. Gegúndez-Arias, and Estefanía Cortés-Ancos. A fast and accurate deep learning method for strawberry instance segmentation. *Computers and Electronics in Agriculture*, 178:105736, 2020.

[5] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.