SEMESTER PROJECT

GrowBotHub
Plant monitoring using computer vision

Student
Aurélien BALICE-DEBBAS

Project coordination
(Computer Vision Lab EPFL)

Mathieu SALZMANN

Joachim HUGONOT

For GrowBotHub coordination
(EPFL-Space Center)

Jean-Paul KNEIB

Alexandre LUCAS

Abstract

The EPFL interdisciplinary project GrowBotHub seeks to monitor plant growth and states of growth. Computer vision applied to the plant is already widely used, but there is no research or dataset on applying those techniques to monitor growth states. This semester project aims to evaluate the relevance of this approach by testing it with an Arabiosis dataset. Through this study, this project aims to give advice on how to create and annotate a good dataset.
.

June 5, 2020

Introduction

Who has never dreamt of going to the moon? But have you thought about how you would eat once arrived? SpaceX Starlink mission reminds us of the importance and unifying power of major space missions.

This is also the vision of ESA Lab, which in 2018 called for the creation of a European inter-university project to simulate a lunar habitat. The Swiss Space Center at EPFL has taken on this mission and that is how Igluna was born.

From this large project, GrowBotHub has emerged as one of the projects offering food production in this lunar habitat. Since 2020, GrowBotHub is also recognized as an interdisciplinary MAKE project at EPFL. The GrowBotHub, aims for a fully automated & autonomous system to grow and pick-up vegetables in a closed-loop fashion. It uses aeroponics (soilless culture), robotics (using an iiwa7 KUKA robot), computer vision and sensors.

This semester project aims to study and develop the computer vision architecture of this interdisciplinary project . A more detailed presentation of GrowBotHub can be found in Appendix A

Plant phonology is the study of plant life cycle in an environment. Computer vision associated to plants, crops and phenotyping is a subject in constant evolution. In a review[23] published in April 2020, well after the beginning of this work, the author presents the current work in the field. Despite its great potential for greenhouse culture, the classification of vegetable growth sages using Deep Learning is almost absent of literature.

However, the study of growth states is a discipline that has long been studied in biology. The most widely used standardized scale today for evaluating plant growth stages is the BBCH scale developed in Germany in 1985 [33, 34]. This norm divide plant growth in 10 different distinguishable states. Specific characteristic of each plant species are described in depth. A summary of the scale and its application to the plants used in this project can be found in Appendix B. This appendix is also used for data annotation.

This work is a preparation work for the capture of a dataset next semester and for the future continuation of this project. The main scope of this project are :

- Define and evaluate an annotation protocol for vegetable growth states
- Test object detection to determine the growth stages of vegetables for multiple growth stages
- Study the performance of the classification and propose tracks to be followed and changes to be made for capturing and annotating images

We will first present the other academic works using Deep Network for plant-related subjects and the datasets available. Then after presenting the data annotation process, we will train à typical neural network for classification and compare the results with ordinal regression built on the same network. We will also train an object detector to evaluate the relevance of the object detection for GrowBotHub. Finally, we will propose some improvements to do to capture a better dataset, for better annotation and the suggest research axes for the continuation of the project.

I. Related work

Over the past decades, most research has emphasized the use of computer vision's techniques applied to agriculture. Most of them use traditional computer vision techniques and manually crafted features . The most significant example would be PlantCV [17] , a dedicated library for plant analysis based on OpenCV.

In recent years, there has been an increasing number of studies using Deep Neural Networks for agriculture and plant-related tasks. We will focus on those recent techniques. Much of the literature falls into four main categories :

- Remote sensing for large crops
- Fine-grained plant and disease classification and pants stress detection
- Plant phenotyping using plant segmentation and leaf counting
- Weed detection

However, there is a relatively small body of literature concerned with the use of computer vision to detect and classify individual plants stages of growth.

A. *Remote sensing*

Computer vision has been widely used for remote sensing in agriculture. Large crops are monitored using satellite imaging or drones. It is a question of studying the same things as we do, but on a macroscopic scale, without necessarily being interested in individual plants. Let's just note a challenge currently running on kaggle on wheat detection[1] linked to ECCV.

B. *Fine-grained plant detection ad stress detection*

Since 2008, the Oxford 102 Flowers Dataset [39] have been used with neural networks for fine-grained classification. At Leaf classification using Deeep Neural Networks [25–27] general plant classification [16] and disease detection on individual plants[10, 38] have also being studied as stress-detection-related fields.

C. *Plant phenotyping and leaf counting*

Recently, Deep learning techniques have been applied for plant segmentation. The Leaf Segmentation Challenge and Leaf Counting Challenge in Computer Vision Problems in Plant Phenotyping Workshop (CVPPP) 2017 [46] and in summer 2020 in ECCV 2020[2] with a new challenge in wheat detection. Most of the recent computer vision techniques are adapted. Plant centre detection [22], leaf counting using a SegNet for segmentation followed by a regression network [3] are recent

[1] www.kaggle.com/c/global-wheat-detection
[2] https://www.plant-phenotyping.org/CVPPP2020

examples. Domain adaptation for leaf segmentation and counting [18] was also used for this topic. In order to make the algorithms more robust, leaf segmentation using synthetic data[52] and adversarial network for data augmentation [57] are also used as wel 3d segmentation for overlapping leaves[29]

In a recent study, a multitask method[12] using a ResNet50 as feature extractor coupled to a task three branches performing simultaneously Leaf counting, leaf projected area and genotype classification gave promising results.

These techniques based on segmentation, volume and leaf count can be used to detect the level of plant maturity. However, these methods are only applicable to leafy plants where maturity for harvest is at stage 4 of the BBC classification, which greatly imitates its use in agriculture excluding fruit vegetables (Tomatoes, Peppers) and root vegetables (carrots).

D. Weed detection

The last category focusses on real time segmentation for weed detection in precision agriculture[35], to reduce the amount of herbicide. Deep learning techniques for weed detection [2, 15, 53, 54] among other plants shows the spectrum of use of this method in environments where it is difficult to identify two different species.

E. Datasets

Several datasets related to plant-related tasks exist.

First let cite the CropDeep Dataset [56] is composed of images from different vegetables, captured in greenhouses and annotated.

Weed detection

Several annotated datasets [8, 41, 50] are available for weed detection and segmentation tasks.

Arabiodopsis

Arabidopsis thaliana is a plant widely used in plant biology for phenotypic and genotypic experiments. It is a member of the mustard family.

Standard dataset of Arabiodopsis and Tobacco used in most research papers and conferences : [36, 47]. A semi annotated tool for plan annotation [37] have also been developed along with these datasets.

Temporal growth of 98 Arabiodopsis plants [49], composed 22 images per plant taken every day forms one of the few dataset available covering several stages of plant growth. Finally, aysntetic generation of Arabiodopsis using adversarial network [19] have been explored by researchers to augment the existing ones.

Even if several datasets exist, the scope of application of the available datasets remains rather limited to certain domains.

F. Face detection

The task of fine-grained plant detection, categorization and stage of growth classification can be related to face/people detection, gender/race [31] [51] categorization and age detection[14, 21, 42]. As shown in the following, we have drawn inspiration from these techniques and it therefore seems important to present the principal ones.

It has been shown that methods based on Ordinal regression for age estimation, and multitask methods with joint age, race ad gender estimation [28, 43, 44] gives better results than separate methods. Debrescu [12] method for plant phenotyping have shown similar results.

In summary, this review showed that neural networks are widely used in plant-associated research, but that, due to the lack of a wide variety of datasets, does not apply to the subject of this project. There is therefore a real lack of studies related to the classification of plants and vegetable growth states.

II. Methods

For this project, we annotate an existing Arabiodopsis dataset with bounding boxes and stages of growth. Using this dataset we trained two classification network, on typical ResNet50 and one version using Ordinal regression. Finally, we trained an object detector to detect plants and classify their stages of growth. Figure 1 shows the different techniques used for this project.

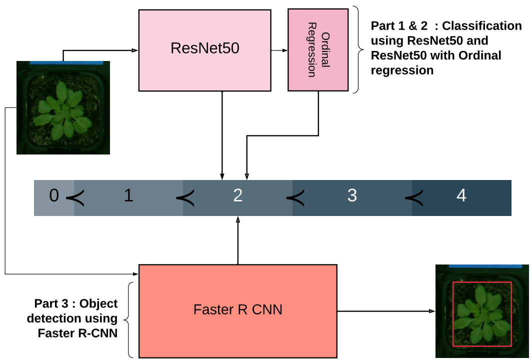


Figure 1: Schematic presentation of the class structure, classification and detection algorithms applied in this project. The classes are not balanced, and each growth phase does not have the same time duration.

A. Dataset and Annotation

As mentioned earlier, no pre-annotated dataset is available. The only dataset available and usable for the classification of vegetable states is the CropDeep Dataset[56] but is unfortunately not available publicly.

The collection of an experimental dataset was initially planned at EPFL this semester with continuous annotation. The COVID situation did not allow to complete it. We finally used this project to find out how to better capture and annotate a dataset for future work.

In order to find a consistent dataset for this pre-study, we contacted some institution:

- Not publicly available dataset :
  - Crop deep dataset[56] : Contacted 2 times, no answer.
  - Australian National University and University of Adelaid. They provided us with a tomato dataset. 40Go, not annotated but not completely relevant for the project.
- Institutions ins Switzerland :
  - Agroscope[3], federal agronomy institute. Contacted 2 distinct research group in charge of digitalization [6] [9] : They don't have any datasets.
  - Lulier, Geneva School of Agronony : They don't have any dataset.
  - Grangeneuve, Fribourg Agronomy Institute : They don't have any dataset but they are open for a collaboration (installing cameras on their greenhouse). Meeting scheduled.

Finally, we used an Arabiodopsis [49] dataset which presents clear images easier to annotate during a continuous period of time. It is composed of 2000 images, representing different Arabiodopsis plants, photographed from the top every day for 22 days. To be used , the dataset has been re-annotated: On the one hand, bounding boxes have been built around each of the plants. On the other hand, each image is associated to its state of growth, going from states 0 to 4, following a slightly modified version of the BBCH scale of Arabiodopsis [4]. A detailed presentation of the annotation process is detailed in Apendix **??**. Note that plant growth is not linear. Each plant takes a different period of time on each stage, Moreover, the first class is underrepresented in the dataset. The capture did not start on the day of seeding. The dataset is unbalanced. Table I shows the number of images per class.

Class	1	2	3	4	5
Number of images	162	462	491	516	431

Table I: Number of annotated images for each class (growth stage)

During the data annotation process, the [55] age data have been used to test the implementation of the classification models.

We divided the project into two different parts, pure classification of plant growth stages and object detection. We tested two methods for classification, a typical classification and ordinal regression. Performing classification - even though our main task also includes detection - and comparing these two methods not only allows us to see which one would be more efficient, but also, depending on the classification results, to gain key insights about the visual representation of plant growth, dependency between classes, boundaries and to be able to adapt the data collection and annotation to improve the final results.

B. Classification

In this part, we propose the method used to classify plants into their stages of growth. We will first present the concept of ordinal regression used to represent the temporal relationship between the different growth stages. We will then show the how we fine-tuned a network to use ordinal regression and the methodology and setup used to compare the network with or without ordinal regression.

1) Ordinal Regression

Ordinal regression is used to assign an ordinal dependency between classes such that for $\boldsymbol{x}_i \in \mathcal{X}$ the ith image and $y_i \in \mathcal{Y} = \{r_1, r_2, ..., r_K\}$ the corresponding rank, the ordinal relationship between images can be written as $r_K \succ r_{K-1} \succ ... \succ r_1$ With images having ordinal discrete dependency,as it is the case for plant growth, this type of classification have shown better results than usual classification methods.

With $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, ordinal regression consists of finding a mapping from images to ranks $h : \mathcal{X} \to \mathcal{Y}$ such as a **loss function** $L(h)$ is minimized.

The usual approach is to perform $K + 1$ binary classifications [30] and adapted it to a Neural Network [40].

In his paper [7] the author proposes a novel approach to ordinal regression addressing some consistency issues encountered with the previous methods, increasing accuracy and reducing the number of parameters at training.

Label Extension and Rank Prediction:

First, we transform an ordinal regression problem of K ranks into $K - 1$ binary classification sub-problems.

$D = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ is extended into $D^k = \{\boldsymbol{x}_i, y_i^k\}_{i=1}^N$ such that $y_i^k = \mathbb{1}\{y_i > r_k\}$.
For example, for $K = 10$ and $y_i = 1$:

$$[2] \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Class importance parameter

In order to address the problems related to class imbalances, we adopt the method described in the papers. Note λ^k the importance parameters for the task. We get :

$$\lambda^k = \frac{\sqrt{M_k}}{\max_{1 \le i \le K-1}\left(\sqrt{M_k}\right)} \tag{1}$$

With S_k the number of examples with rank superior to r_k and $M_k = \max(S_k, N - S_k)$

Network layer and Loss

The output of the fully connected layer of the backbone network is $h(\boldsymbol{x}_i, \boldsymbol{W})$. With $\boldsymbol{W}$ a weight matrix.

Consider the logistic sigmoid function $s(z) = \frac{1}{1+\exp -z}$.

[3] https://www.agroscope.admin.ch/agroscope/fr/home.html

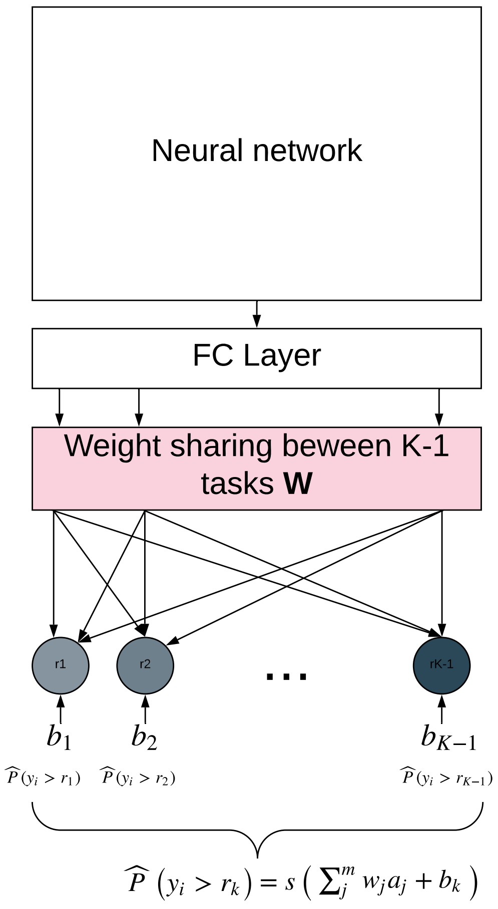


Figure 2: The Ordinal regression pipeline.

The predicted probability for each class is then :

$$\widehat{P}\left(y_i^k = 1\right) = \widehat{P} = s(h(\boldsymbol{x}_i, \boldsymbol{W}) + b_k) \tag{2}$$

hence, as shown in Figure 2

$$\widehat{P}\left(y_i > r_k\right) = s\left(\sum_j^m w_j a_j + b_k\right) \tag{3}$$

The loss function to minimize is then a weighted cross entropy loss:

$$L(W, b) = -\sum_{i=1}^{N}\sum_{k=1}^{k-1} \lambda^k \left[log\left(s\left(h\left(x_i, W\right) + b_k\right)\right) y_i^k + \log\left(1 - s\left(h\left(x_i, W\right) + b_k\right)\right)\left(1 - y_i^k\right)\right] \tag{4}$$

For each rank, the binary labels are obtained using :

$$f_k\left(x_i\right) = \mathbb{1}\left\{\widehat{P}\left(y_i^k = 1\right) > 0.5\right\} \tag{5}$$

such that $f_1\left(x_i\right) \geq f_2\left(x_i\right) \geq ... \geq f_{K-1}\left(x_i\right)$

Evaluation

We use a Cumulative Score (CS) as an evaluation metric. It gives the proportions of images where difference between the predicted labels and the real label are under a threshold T. using this metric with a threshold $T = 0$ gives the accuracy.

$$CS\left(t\right) = \frac{1}{N}\sum_{i=1}^{N} 1\left\{\left|y_i - \left(x_i\right)\right| \leq T\right\}$$

We also compute the confusion matrix, precision and recall.

Number of parameters

For a network with m output nodes in the last fully connected layers, the ordinal regression algorithm used in this experiment shares weight between the tasks and uses one additional neuron for each. The number of parameters is then $m + K - 1$

2) Network

We use a ResNet50[20] network pretrained on ImageNet [11].

Deep Residual Neural networks have gained popularity due to their use of skip connection, making the availability for the network to have variable size solving the problem of vanishing gradient and allowing the network to be deeper. The vanishing gradient problem occurs when a network is to deep : up to a certain point in back propagation, the gradient can become very small. Skip connections allow shortcuts for the gradient to flow through.

ResNet50 has 50 layers, which makes it a good network to test our methods with a relatively small training time. There exist retrained ResNet on different networks, which makes it easy to fine-tune. It is a network that's already been used for phenotyping tasks on Arabiodopsis.[13]

We modify the reference ResNet50 keeping the retrained weights. We removed the last layer, made to classify 100 classes and replaced it with 5 classes corresponding to our stages of growth. In the Ordinal version, we replaced it with the ordinal head.

3) Training

To compare typical classification and classification using ordinal regression, we generate 5 random data splits for training, evaluation and test. Each random split divide the dataset with proportion of 60% for training and 20% for validation and test. We then trained each split separately.

Data augmentation

Data augmentation is usually used it to increase the size of available datasets, increase diversity of example and making the network more robust. As we test here the relevance of the models, assessing if they are suitable for the task and understanding data annotation and the stricture of the data, we keep data augmentation to a minimum. We first resize the images to 224px time 224px. We then, at training perform a random crop. A validation, a random centre crop. We then normalize the images, using the Image net standard normalization values.

Class balancing

To balance the classes and perform a comparison between the two methods, we computed the class balancing parameters according the the ordinal regression equation (Equation (1)).

Parameters

The model was trained on Google Collab using an NVIDIA T4 GPU. We used Stochastic Gradient Descent with a learning rate of 0.005, a momentum of 0.9 and weight decay of 0.0005 We trained the model for 40 epoch and decrease the learning rate by a factor 0.1 every 15 epoch.

For both ordinal typical ResNet and ordinal regression, we use a Cross entropy loss (4), which is the common loss used for multiclas classification and allows weighting.

At evaluation we use accuracy as a metric.

Cropping

In order to have an overview of the results in terms classification for the object detection task, we also trained each model with the images cropped in the bounding box area. The same splits, augmentation and parameters are used.

C. Detection

To perform plant detection, we fine-tuned a faster R-CNN [45] network with a ResNet50 retrained backbone used as a feature extractor. Using RN as backbone is consistent with the previous section. One evaluation metric for the boxes proposed by Faster R-CNN is Intersection over Union (IoU) which computes the ratio between the area of the intersection of two boxes and the area of the union of both boxes.

Faster R-CNN uses a Region Proposal Network, initialized by the backbone, which generates region proposals.

The loss function is the sum of both log loss function from classification between two classes and boxes $\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}}$.

Parameters

We use a pretraiend network on Coco datasets [32] and fine-tuned it. We used a Stochastic Gradient Descent, with a learning rate of 0.005, momentum of 0.9 and decay of 0.0005. We trained for 20 epoch and decrease the learning rate of a factor 0.1 every 3 epoch. We chose to stay as close as possible from the parameters used for classification.

We used the already implemented pytorch object detection reference training script[4] based on a COCO data structure.

III. Results

In this section, we will present the results for ResNet50 classification, ResNet50 with ordinal regression and Faster R-CNN detection. We also presents the classification algorithm trained on images cropped to the bounding boxes, and broader misclassification analysis.

[4]https://github.com/pytorch/vision/tree/master/references/detection

A. Classification

We finetuned both ResNet50 and ResNet50 with Ordinal regression regression five time with the different splits. We the evaluate each splits using a confusion matrix. Using this confusion matrix, we compute average accuracy, precision and recall.

Table II presents the averaged accuracy for the five splits. We could have expected that the Ordinal regression regression would have better results than the typical classification but it is only the case for the 5^{th} split. By averaging the tests results, we get an accuracy of 70% for the typical ResNet50 classification and an accuracy of 68% for ordianl regression for the test sets. According to the original ResNet paper[20], a RN50 trained on image net have a top-1 accuracy of 77%.

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
RN Accuracy	0.7094	0.7215	0.6973	0.7022	0.6877
RNO Accuracy	0.6659	0.6852	0.6780	0.6634	0.7094

Table II: Comparison of the accuracy of a Reset-50 network, with (RNO) and without Ordinal regression regression (RN), on 5 different splits of the images.

Looking at each classes individually, Table III shows the precision and recall, for both RN and RNO for each random splits and the result in average. We first notice that for almost all splits, the best recall for both ResNet50 and Ordinal regression is reached for class 4 with, respectively an average of 86% and 80%. This class also the one with the best precision, with respectively 82% and 86%. Class 4 is directly followed by Class 1 with an average score of 67% and 71%.

On the other hand, the other 3 classes share a lower recall. Depending on the splits, it is not always the same class that has the lowest score. In average, the worst recall for both RN and RNO is reached for class 0, with 65% and 57% on average. However, worst precision of 54% is achive ar class 0 for ResNet50 but for class 3 for Ordinal regression, with 58%

Figure 3b (3a and 3a) shows the averaged confusion matrix. The detailled confusion matrices can be found on Appendix D

From the confusion matrices, we first see that the classification errors are mostly done on neighboring classes. The most significant error of classification is images belonging to class zero being classified with label 1. It corresponds in average to 33% and 43% respectively for Resnet50 and ResNet50 with ordinal regression.

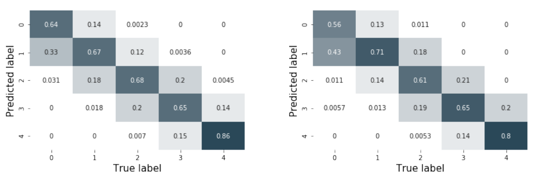
Cropped images

In order to give insight on the classification results for the object detector, we trained both the ResNet and the Ordinal regression ResNet 5 splits on the cropped images. We used the same parameters as in the previous section.

The results averaged accuracy obtained with this method are shown in Table IV. We note a decrease

Class		0	1	2	3	4
Seed 1						
ResNet	Precision	0.487	0.750	0.642	0.711	0.818
	Recall	0.643	0.657	0.709	0.640	0.862
RNO	Precision	0.378	0.674	0.663	0.622	0.808
	Recall	0.636	0.652	0.594	0.609	0.816
Seed 2						
ResNet	Precision	0.677	0.744	0.606	0.721	0.859
	Recall	0.636	0.693	0.708	0.672	0.869
RNO	Precision	0.839	0.573	0.683	0.577	0.882
	Recall	0.531	0.691	0.602	0.711	0.852
Seed 3						
ResNet	Precision	0.514	0.833	0.517	0.725	0.775
	Recall	0.643	0.686	0.625	0.627	0.894
RNO	Precision	0.600	0.643	0.644	0.605	0.847
	Recall	0.600	0.720	0.609	0.641	0.769
Seed 4						
ResNet	Precision	0.560	0.766	0.605	0.700	0.791
	Recall	0.560	0.661	0.714	0.638	0.868
RNO	Precision	0.760	0.543	0.720	0.469	0.901
	Recall	0.452	0.718	0.652	0.625	0.746
Seed 5						
ResNet	Precision	0.462	0.820	0.534	0.625	0.868
	Recall	0.750	0.664	0.611	0.663	0.798
RNO	Precision	0.692	0.562	0.777	0.644	0.857
	Recall	0.643	0.794	0.611	0.684	0.839
Average						
ResNet	Precision	0.540	0.783	0.581	0.696	0.822
	Recall	0.646	0.672	0.674	0.648	0.858
RNO	Precision	0.654	0.599	0.697	0.584	0.859
	Recall	0.573	0.715	0.614	0.654	0.804

Table III: Precision and recall for the 5 random splits, for typical ResNet50 classification and for classification using a ResNet50 with ordinal regression (RNO). Best results are shown in green, worst in red.



(a) ResNet50 average *(b) Ordinal regression average*

Figure 3: Confusion matrices for a ResNet50 with and without Oridnal regression finetuned on the plant dataset for classification. The data was split randomly 5 times.

2% for RN, reaching 68.5% of accuracy and a decrease of2.5% with average accuracy of 64.6%.

Average precision and recall are presented in Table V. In the same manner than for accuracy, we get a decrease in both precision and recall compared to previous section. Averaged confusion matrix is presented in Figure 4 and detailed confusion matrices for each split is shown in Appendix D

Annotation errors

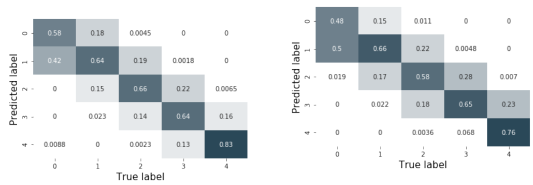
In order to better understand the nature of the images that are not classified correctly, we analyzed their position within each classes. For each images, we determines if they belong to the border of a class, regardless of whether it is an upper or lower border. We neglect both

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
ResNet50 cropping accuracy	0.656	0.685	0.668	0.700	0.714
RNO cropping accuracy	0.622	0.632	0.644	0.678	0.654

Table IV: Accuracy comparison for 5 different splits, after cropping the images at the bounding boxes dimension. For ResNet50 and ResNet50 with ordinal regression (RNO)

Class		0	1	2	3	4
Average						
ResNet	Precision	0.440	0.696	0.587	0.711	0.833
	Recall	0.614	0.642	0.657	0.64	0.834
RNO	Precision	0.665	0.483	0.641	0.525	0.935
	Recall	0.480	0.668	0.581	0.647	0.761

Table V: Precision and recall for the 5 random splits, for typical ResNet50 classification and for ResNet with ordinal regression (RNO).



(a) ResNet50 for crop image *(b) ResNet with Ordinal regression for crop image*

Figure 4: Averaged confusion matrix for both ResNet50 and ResNet50 with ordinal regression

extremes. We perform his estimate for the ResNet50 which previously gave us the best results.

For each classes, we computed the proportion of images belonging to the border, and compere it to the proportion of misclassified images belonging to the border. The detailed results are presented in Table VI

In average, 33% of the test set images belongs to borders. This results jumps to 50% for misclassidied images. We note a increas of 17% in average for the misclassified images; performing a z test Taking each classes individually, in averag class 1, 2 and 3 have the biggest proportion of border images, which supports the fact that class 0 and 4 have only one edge.

However, the difference in proportion between misclassified images and the number of images shows more significant differences between classes. Classes 1,2 and 3 have a difference of 17.5%, 14.1% and 12.6% respectively whereas class 0 reaches an difference of 19% and class 4 a difference of 23.4%

On average, the Neural network has more difficulty classifying images that are on the edge of the class.

Moreover, even if the algorithm is more accurate on class 4, it is also the class that has the highest number of images on the edges that are not well classified.

Class	0	1	2	3	4
Random split 1					
Misclassified	0.272	0.629	0.428	0.548	0.368
Border	0.25	0.364	0.350	0.360	0.171
Average misclassified : 0.486					
Average : 0.305					
Seed 2					
Misclassified	0.545	0.558	0.607	0.472	0.363
Border	0.333	0.356	0.419	0.396	0.144
Average misclassified : 0.525					
Average : 0.344					
Seed 3					
Misclassified	0.307	0.5	0.6	0.363	0.444
Border	0.25	0.347	0.337	0.324	0.197
Average misclassified :0.454					
Average :0.305					
Seed 4					
Misclassified	0.5	0.55	0.444	0.6	0.6
Border	0.266	0.389	0.318	0.370	0.253
Average misclassified : 0.542					
Average :0.337					
Seed 5					
Misclassified	0.666	0.452	0.464	0.512	0.428
Border	0.241	0.357	0.411	0.415	0.271
Average misclassified: 0.481					
Average :0.361					
Average					
Misclassified	0.458	0.538	0.508	0.499	0.441
Border	0.268	0.363	0.367	0.373	0.207
Average misclassified :0.497					
Average : 0.330					

Table VI: Comparison of images belonging to borders, for the whole test set and for the miscassified images.

B. Detection

Bounding boxes

On the test dataset, the Faster R-CNN method gives an average result of intersection over union of 0.935. Which is an excelent score.

In term of classification, we obtain an average accuracy of 0.6869 which is comparable to the ResNet from previous section.

Class	0	1	2	3	4
Detection					
Precision	0	0.826	0.674	0.678	0.828
Recall	0	0.618	0.653	0.642	0.854
Classification and crop					
Precision	0.243	0.707	0.537	0.689	0.849
Recall	0.818	0.602	0.586	0.591	0.824
Classification					
Precision	0.487	0.750	0.642	0.711	0.818
Recall	0.643	0.657	0.709	0.640	0.862

Table VII: Precision and recall for detection, with as references the classification from previous section, for the same random split

Results based on the first split with the first seed

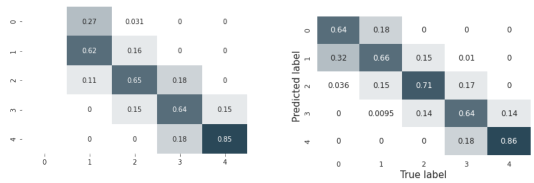
Classification

A comparaison with the ResNet50 classification with and without cropping is presented in Table VIII.

From the precision and recall (Table VIII, and the confusion matrix (Figure 5, we notice that the detection

Class	0	1	2	3	4
IoU	0.	0.937	0.943	0.943	0.928

Table VIII: Intersection over Union scores for each of the 5 classes, using Faster R-CNN



(a) Faster R-CNN object detection *(b) ResNet50 classification*

Figure 5: Confusion matrix for Faster R-CNN detection compared to classification using a ResNet50

do not classify images for the fist class. However, the classifier performs better than the network with cropping for all the other classes, and slightly (1% in recall) worst than the ResNet50 classification.

IV. Discussion

The results of the previous section are encouraging, whether it is for classification or plant detection. The ordinal classification, which seemed to be promising, finally gave less good results than the classical classification. Let's first note that we didn't use state-of-the-art techniques neither for classification nor for detection.

Results about wrongly classified data, however, seems to reveal issues linked to annotation and data structure.

In the dataset we used, there is a relatively small number of consecutive images within each class. On average there are 4 images per class. Of course, this number varies greatly depending on the class, but 1 image out of 3 is on a transition between classes (excluding extreme classes which only have one neighbour). This makes class transition problems steeper.

As 50% of the images that were not classified correctly belongs to neighbouring classes As shown when studying the class transitions, the algorithm, have more difficulties classifying images in transition. We therefore propose some solutions to try to address those issues for the following of the project :

1) Increase the number of images taken each day. This will reduce the number of neighbouring classes, makes the annotation for each class easier as the transitions will be more precise and more visible. Jumping from 1 image per day to 3 to 4 images per day seems a reasonable increase. However, the balancing Day/Night could cause issues in balancing of the dataset.
2) With more images, smooth labels can be envisaged as we are dealing with imprecise transitions.

In addition, having annotated the images myself, I realized that the transition between classes is generally difficult to perceive. Reducing the number of stages, or only taking relevant one can also help.

When annotating the data, as a human, this last class was the "easiest" to classify. The human recognizable features were clear (element at the centre of the plant). The high number of misclassified images, however, could be explained by the fact that, again, some precise key images for class transition where difficult to get. The first stage of growth, however, was the more difficult to classify, because of the different development stages of the plants.

Another element that may have influenced the problems of classification is the diversity of species. Indeed, this dataset contained several species of the same plant, with significant variations in the stages of development. Both in the size and shape of the leaves and the speed of development. These differences were problematic during annotation.

Conclusion

This project shows have shown that good results can be obtained on the classification of plants according to their state of growth as well as their detection.

It also raises issues related to these methods, annotation errors and data issues.

This project is the basis for the further development of GrowBotHub, either in the realization of a dataset or in the development of other networks for classification and detection.

Next steps

What	When	How / Who
Capture a dataset for various vegetable	Jun. 2020 - Dec. 2020	Grenhouse (Under dicussion) Aurelien
Annotation	Oct. 2020 - Feb 2021	Lead : Aurelien Other : GBH IC Team
Detection pipeline	Spring 2021	1 or 2 Semester projects

We also realized a guide for the realization of the dataset and the annotation for the use of GrowBotHub. It gives practical advice, based on the experience and lessons learned from this project, and addresses some questions to be solved.

https://docs.google.com/document/d/150cxHsH3zntnh6agdZ--wUrVjlyMC1lq91eaUDO1258/edit?usp=sharing

References

[1] EPPO (2020) EPPO Global Database (available online).

[2] Alwaseela Abdalla, Haiyan Cen, Liang Wan, Reem Rashid, Haiyong Weng, Weijun Zhou, and Yong He. Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. *Computers and Electronics in Agriculture*, 167(December):105091, 2019.

[3] Shubhra Aich and Ian Stavness. Leaf counting with deep convolutional and deconvolutional networks. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-Janua:2080–2089, 2017.

[4] D. C. Boyes, A. M. Zayed, R. Ascenzi, A. J. McCaskill, N. E. Hoffman, K. R. Davis, and J. Görlach. Growth stage-based phenotypic analysis of Arabidopsis: A model for high throughput functional genomics in plants. *Plant Cell*, 13(7):1499–1510, 2001.

[5] I. G. Burns. Influence of plant nutrient concentration on growth rate: Use of a nutrient interruption technique to determine critical concentrations of N, P and K in young plants. *Plant and Soil*, 142(2):221–233, 1992.

[6] C Camps and D Tran. Digitalisation en serre. 2018.

[7] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank-consistent Ordinal Regression for Neural Networks. 2019.

[8] Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter, Wolfram Burgard, and Cyrill Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *International Journal of Robotics Research*, 36(10):1045–1052, 2017.

[9] R. Steiner C. Bucher H. Möri T. Wyssa F. Duckert S. Hauenstein R. Matter T. Anken R. Total M. Keller D. Hodel, L. Collet. Application ciblée par plante, sur culture de Pak-Choi, à l'aide du robot pulvérisateur Steketee. Technical report, Agroscope, 2020.

[10] Chad DeChant, Tyr Wiesner-Hanks, Siyuan Chen, Ethan L Stewart, Jason Yosinski, Michael A Gore, Rebecca J Nelson, and Hod Lipson. Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning. *Phytopathology*, 107(11):1426–1432, 11 2017.

[11] J Deng, W Dong, R Socher, L.-J. Li, K Li, and L Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[12] Andrei Dobrescu, Mario Valerio Giuffrida, and Sotirios A. Tsaftaris. Doing More With Less: A Multitask Deep Learning Approach in Plant Phenotyping. *Frontiers in Plant Science*, 11(February):1–11, 2020.

[13] Andrei Dobrescu, Livia C.T. Scorza, Sotirios A. Tsaftaris, and Alistair J. McCormick. A "Do-It-Yourself" phenotyping system: Measuring growth and morphology throughout the diel cycle in rosette shaped plants. *Plant Methods*, 13(1):1–12, 2017.

[14] Yuan Dong, Yinan Liu, and Shiguo Lian. Automatic age estimation based on deep learning algorithm. *Neurocomputing*, 187:4–10, 2016.

[15] Mads Dyrmann. Automatic Detection and Classification of Weed Seedlings under Natural Light Conditions. 2017.

[16] Mads Dyrmann, Henrik Karstoft, and Henrik Skov Midtiby. Plant species classification using deep

convolutional neural network. *Biosystems Engineering*, 151(2005):72–80, 2016.

[17] Noah Fahlgren, Maximilian Feldman, Malia A. Gehan, Melinda S. Wilson, Christine Shyu, Douglas W. Bryant, Steven T. Hill, Colton J. McEntee, Sankalpi N. Warnasooriya, Indrajit Kumar, Tracy Ficor, Stephanie Turnipseed, Kerrigan B. Gilbert, Thomas P. Brutnell, James C. Carrington, Todd C. Mockler, and Ivan Baxter. A versatile phenotyping system and analytics platform reveals diverse temporal responses to water availability in Setaria. *Molecular Plant*, 8(10):1520–1535, 2015.

[18] Mario Valerio Giuffrida, Andrei Dobrescu, Peter Doerner, and Sotirios A. Tsaftaris. Leaf counting without annotations using adversarial unsupervised domain adaptation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019-June:2590–2599, 2019.

[19] Mario Valerio Giuffrida, Hanno Scharr, and Sotirios A. Tsaftaris. ARIGAN: Synthetic arabidopsis plants using generative adversarial network. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-Janua(i):2064–2071, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, 2016.

[21] Z Hu, Y Wen, J Wang, M Wang, R Hong, and S Yan. Facial Age Estimation With Age Difference. *IEEE Transactions on Image Processing*, 26(7):3087–3097, 2017.

[22] Yotam Itzhaky, Guy Farjon, Faina Khoroshevsky, Alon Shpigler, and Aharon Bar Hillel. Leaf counting: Multiple scale regression and detection using deep CNNs. *British Machine Vision Conference 2018, BMVC 2018*, pages 1–13, 2019.

[23] Yu Jiang and Changying Li. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant Phenomics*, 2020:1–22, 2020.

[24] Ju Yeon Lee, Arifur Rahman, Hossain Azam, Hyung Seok Kim, and Man Jae Kwon. Characterizing nutrient uptake kinetics for efficient crop production during Solanum lycopersicum var. Cerasiforme Alef. Growth in a closed indoor hydroponic system. *PLoS ONE*, 12(5):1–21, 2017.

[25] S H Lee, C S Chan, P Wilkin, and P Remagnino. Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 452–456, 2015.

[26] Sue Han Lee, Chee Seng Chan, Simon Joseph Mayo, and Paolo Remagnino. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, 71:1–13, 2017.

[27] Sue Han Lee, Yang Loong Chang, Chee Seng Chan, and Paolo Remagnino. Plant identification system based on a convolutional neural network for the lifeclef 2016 plant classification task. *CEUR Workshop Proceedings*, 1609:502–510, 2016.

[28] Gil Levi and Tal Hassncer. Age and gender classification using convolutional neural networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2015-Octob:34–42, 2015.

[29] Dawei Li, Yan Cao, Guoliang Shi, Xin Cai, Yang Chen, Sifan Wang, and Siyuan Yan. An Overlapping-Free Leaf Segmentation Method for Plant Point Clouds. *IEEE Access*, 7:129054–129070, 2019.

[30] Ling Li and Hsuan Tien Lin. Ordinal regression by extended binary classification. *Advances in Neural Information Processing Systems*, pages 865–872, 2007.

[31] Xinyue Li and Samuel Cheng. Pedestrian Gender Detection Based on Mask R-CNN. pages 2082–2086, 2020.

[32] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755, 2014.

[33] D Lorenz, K Eichhorn, H Bleiholder, R Klose, U Meier, and E Weber. Phänologische Entwicklungsstadien der Rebe (Vitis vinifera L. ssp. vinifera). – Codierung und Beschreibung nach der erweiterten BBCH-Skala. *Vitic. Enol. Sci.*, 49(2):66–70, 1994.

[34] Uwe Meier. *BBCH BBCH Monograph*. 1997.

[35] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2229–2235, 2018.

[36] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*, 81:80–89, 2016.

[37] Massimo Minervini, Mario Valerio Giuffrida, and Sotirios Tsaftaris. An interactive tool for semi-automated leaf annotation. pages 1–6, 2015.

[38] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7(September):1–10, 2016.

[39] Maria Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008*, pages 722–729, 2008.

[40] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output CNN for age estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:4920–4928, 2016.

[41] Alex Olsen, Dmitry A Konovalov, Bronson Philippa, Peter Ridd, Jake C Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, Brendan Calvert, Mostafa Rahimi Azghadi, and Ronald D White. Deep-Weeds: A Multiclass Weed Species Image Dataset for Deep Learning. *Scientific Reports*, 9(1), 2019.

[42] Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model. (1), 2017.

[43] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2019.

[44] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. An All-In-One Convolutional Neural Network for Face Analysis. *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, pages 17–24, 2017.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[46] H. Scharr, T. Pridmore, and S. A. Tsaftaris. Computer Vision Problems in Plant Phenotyping, CVPPP 2017: Introduction to the CVPPP 2017 Workshop Papers. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-Janua:2020–2021, 2017.

[47] Hanno Scharr, Massimo Minervini, Andreas Fischbach, and Sotirios A. Tsaftaris. Annotated image datasets of rosette plants. *Technical Report No. FZJ-2014-03837*, (July):1–16, 2014.

[48] Aaron Steil. Vegetable Harvest Guide. *Horticulture and Home Pest News*, 2004.

[49] Sarah Taghavi Namin, Mohammad Esmaeilzadeh, Mohammad Najafi, Tim B. Brown, and Justin O. Borevitz. Deep phenotyping: Deep learning for temporal phenotype/genotype classification. *Plant Methods*, 14(1):1–14, 2018.

[50] Dimitar Trajanov and Verica Bakeva. ICT Innovations 2017 Data-Driven Innovation. 778(September):265–273, 2017.

[51] Thanh Vo, Trang Nguyen, and C. T. Le. Race recognition using deep convolutional neural networks. *Symmetry*, 10(11):1–15, 2018.

[52] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. *British Machine Vision Conference 2018, BMVC 2018*, 2019.

[53] Jialin Yu, Arnold W. Schumann, Zhe Cao, Shaun M. Sharpe, and Nathan S. Boyd. Weed Detection in Perennial Ryegrass With Deep Learning Convolutional Neural Network. *Frontiers in Plant Science*, 10(October):1–9, 2019.

[54] Jialin Yu, Shaun M. Sharpe, Arnold W. Schumann, and Nathan S. Boyd. Deep learning for image-based weed detection in turfgrass. *European Journal of Agronomy*, 104(October 2018):78–84, 2019.

[55] Song Yang Zhang Zhifei and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[56] Yang Yang Zheng, Jian Lei Kong, Xue Bo Jin, Xiao Yi Wang, Ting Li Su, and Min Zuo. Cropdeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors (Switzerland)*, 19(5), 2019.

[57] Yezi Zhu, Marc Aoun, Marcel Krijn, and Joaquin Vanschoren. Data Augmentation using Conditional Generative Adversarial Networks for Leaf Counting in Arabidopsis Plants. *Bmvc*, pages 121–125, 2018.

Appendix

A. *GrowBotHub*

This Chapter first presents the general setup and System of the GrowBotHub project in order to frame the computer vision task. It then presents and discuss the different approaches possible in computer vision. Finally, it makes recommendations for future improvements.

System setup

As stated in the introduction, this semester project is part of a wider project : GrowBotHub. In order to better understand the objectives of this research, it is necessary to understand the global focus and the design of certain sub-systems that constitute constraints in the implementation of vision algorithms.

This project implements the camera and vision subsystem.

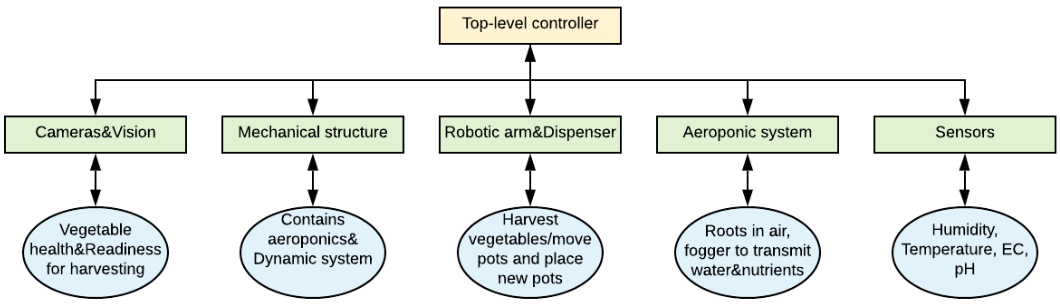


Figure 6: GrowBotHub sub-systems

Subsystems

The structure consist of different main parts listed in the table IX bellow. The different parts can also be shown in Figure 7.

Name	Description
Carousel	The GBH structure consisting of 4 shelves rotating around a axis
Shelf	Aeroponic self, where the vegetables grow. Each Shelf contains 16 plants
Working shelf	Only one shelf can be accessed by the robotic arm to pick vegetable. **??**
Working position	The carousel can be put in ”working position” when the working shelf is on top as in Figure 7. Whis position is reached when pots needs to be picked up and moved.
Day shelves	Because of the rotation of the carousel, only two shelves are visible by the camera and sunlit by the lamps. Figure
Day position	Position of the carousel when the thow ”day shelves” are visible and at the same level. As shown in Figure

Table IX: The different parts of the GrowBotHub System

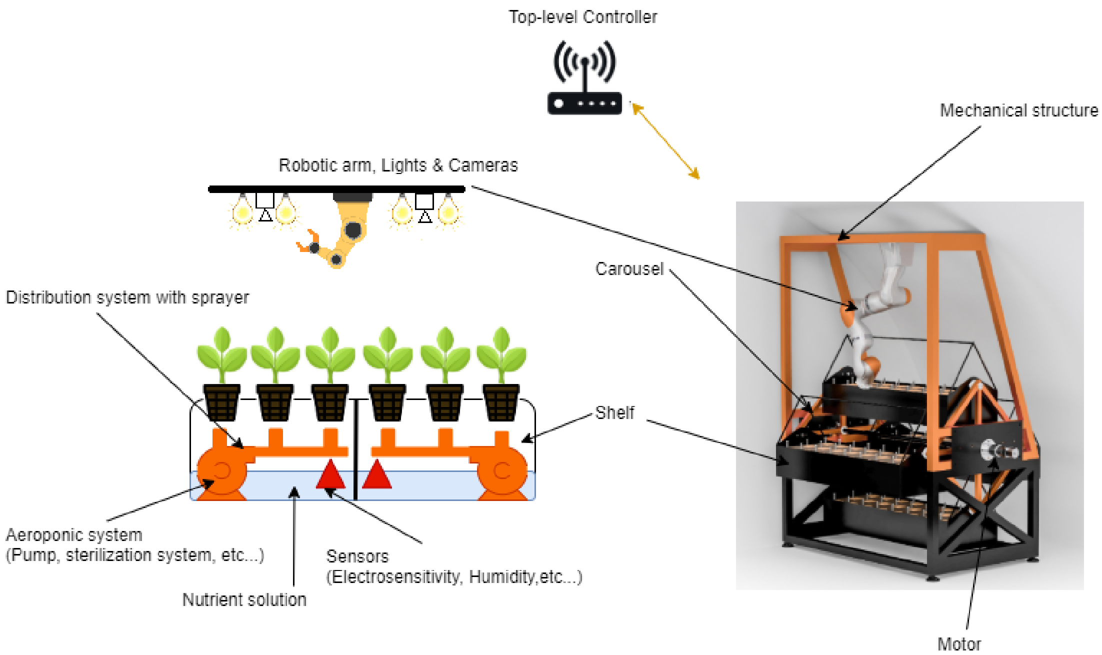


Figure 7: GrowBotHub sub-systems

Vegetable

Aeroponics is a soil less culture. Water mixed with nutrients is sprayed directly on the roots of the plants. Each shelf have two Aeroponic systems. The concentration of nutrients in one Aeroponic subsystem is the same, but at system level, each shelf have a different concentration of nutrient.

Research have shown that varying the concentration of nutrients according to the growth stages of vegetable can influence and optimize the growth[5][24]. The GrowBotHub project takes this aspect of optimization into account : plants are moved (using the robotic arm) according to their stages of growth in different shelves containing distinct level of nutrients.

Vegetable choice :

For this project, six vegetables are used. The vegetable are listed in Table X as well as their growing time[48] and species [1].

Name	Scientific name	Specie	BBCH name	Growth time
Strawberries	Fragaria x ananassa (FRAAN)	Rosaceae (1ROSF)	Fragaria x ananassa (FRAAN)	28-30
Peppers	Capsicum (1CPSG)	Solanales (1SOLO)	Solaneous fruit	60-90
Radishes	Raphanus sativus (RAPSR)	Brassicales (1CAPO)	Root and stem vegetable	25-40
Rocket	Eruca vesicaria (ERUVV)	Brassicales (1CAPO)	Root and stem vegetable	
Spinach	Spinacia (1SPQG)	Amaranthaceae (1AMAF)	Leafy vegetables not forming heads	45-60
Lettuce	Lactuca sativa (LACSA)	Asteraceae (1COMF)	Leafy vegetables forming heads	45-60

Table X: Vegetable used in the GrowBotHub project

According to the BBCH scale, the chosen vegetable are ready to be harvested at different states depending on their species. Leafy vegetables are ready at the end of stage 4 while fruit vegetables are ready at stage 8.

The table below shows the harvesting states for the different vegetables.

	Stage	Strawberries	Peppers	Radishes	Rocket	Spinach	Lettuces
0	Germination, sprouting, bud development	Not ready	Not ready	Not ready	Not ready	Not ready	Not ready
1	Leaf development	Not ready	Not ready	Not ready	Not ready	Not ready	Not ready
2	Formation of side shoots, tillering	Not ready	Not ready	Not ready	Not ready	Not ready	Not ready
3	Stem elongation or rosette growth, shoot development	Not ready	Not ready	Not ready	Not ready	Not ready	Not ready
4	Development of harvestable vegetative plant parts, bolting	Not ready	Not ready	Harvest (end of)	Harvest (end of)	Harvest (end of)	Harvest (end of)
5	Inflorescene emergence, heading	Not ready	Not ready	To late	To late	To late	To late
6	Flowering	Not ready	Not ready	To late	To late	To late	To late
7	Development of fruit	Not ready	Not ready	To late	To late	To late	To late
8	Ripening or maturity of fruit and seed	Harvest (end of)	Harvest (end of)	To late	To late	To late	To late
9	Senescence, beginning of dormancy	To late	To late	To late	To late	To late	To late

Table XI: Stages of growth of the vegetable

In addition to what has been mentioned above, 3 other peculiarities of the system are to be noted :

Different seeding times

In order to have a continuous food production, the vegetables are seeded at different times.

Day and night cycle

The carouse is rotating twice a day alternating the shelves in day position.

Each plant have an unique position and an unique stage of growth :

Due to the different growth time, seeding time and nutrient concentration optimization; each plant in the system have different stage of growth. Moreover, plant position (between shelves and in each shelf) vary during time .

Information available

At each instant the position of the vegetable on the shelves are supposed to be known, as well as an history of the previous states and positions.

However, the robot moving the vegetables being blind, the system have no confirmation that a plant was moved by the robot.

Objectives

The study objectives are based on requirements taking into account the main function, constraints and interfaces. To these objectives, it is necessary to add Computer Vision specific constraints which are the creation of the dataset, the annotation process and the ease of implementation.

Requirements

The above requirements (Table XIII) constitute a base of criteria used in the analysis and choice of vision algorithms adapted to the system.

Number	Type	Requirement	Verification
R1	Functional requirement	The system shall classify plants state of growth	Classification metrics
R2	Constrains requirement	The system shall not require require a long calibration phase before usage	
R3	Interface requirement	The system shall be integrated into the overall structure	
R4	Performance requirements	The system shall be embedded	
R5	Functional requirements	The system should verify that the robot takes a plant	

Table XII: Main requirements of the subsystem

B. BBCH Scale and growth stages

Arabidopsis Growth Stages

One of the available dataset consist of sequences of Arabidopsis. In order to use it, the data is annotated using BBCH Scale [4].

Literature gives description for Growth stage 1, 3, and 5. Those stages are overlaping and clear definitions of tuningpoints is hard. The stages 2 : ”Formation of sideshoots, tillering” and Stage 4 are not present in the scale. Thise stages are induced by the overlaping parts in the scale. Annotating those stages is crutial in the consistency of our algorithm. The classification is detailed in Figure8.

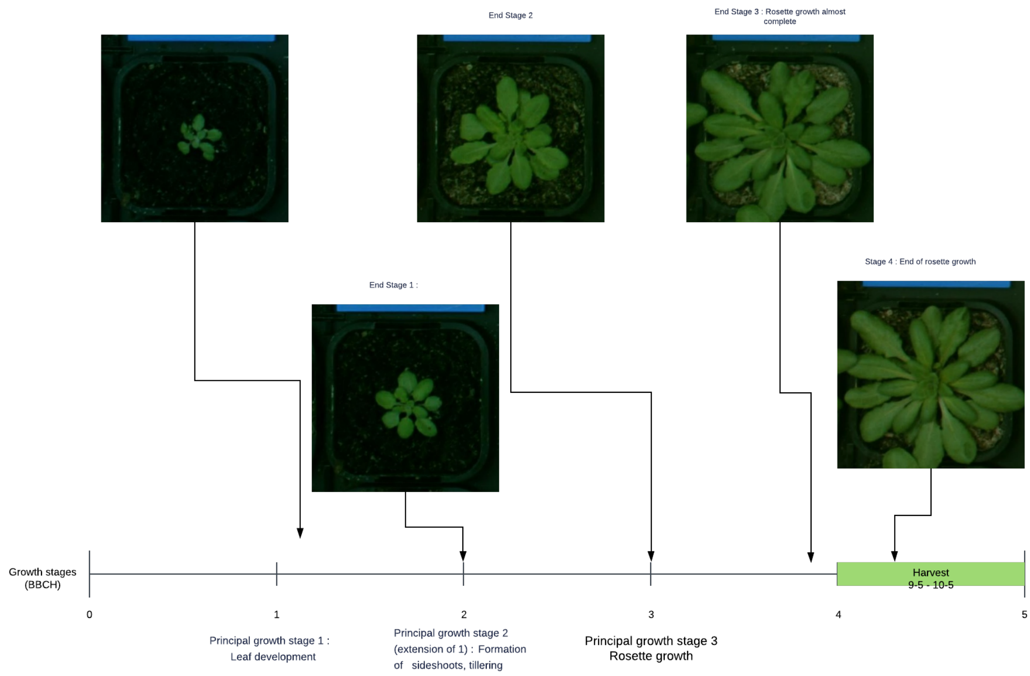


Figure 8: Growth stage for annotation of Arabiosis

Arabidopsis growth stages are close to Rocket and Spinach’s. The scale of annotation can be adapted.

C. Timeline and assesment

Due to the cahotic aspect of this semester, I would like to summarize the different stages of the project's realization.

Timeline

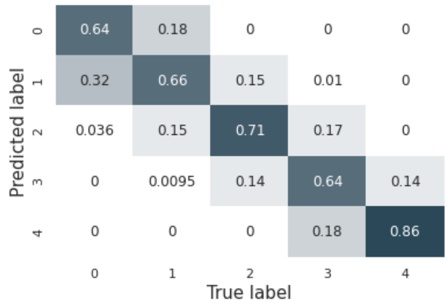
Semester Week	Event	Description
1	Normal week	Preliminary research 2
Normal week	First system pipeline and lliterature	
3	Normal week	Implemenation of binary Object detection and start implementation of segmentation (Last semester repport)
4	Pre COVID, mid-term event Igluna	Firt, we had the mid-term event for the interdisciplinary project (planed in advance) so no real work except synchronisation with the other teams. Moreover, as EPFL has announced a closing of the campus, we have tried in urgency (Friday, Saturday, Sunday) to secure as much as possible the growth of vegetables in order to have datasets. We hoped to be able to come from time to time to adjust the nutrients. This was not possible, so we lost our datasets.
5	Covid- No project	EPFL has imposed the end of semester projects.
6	Covid - No project	EPFL has imposed the end of semester projects.
7	Hackthon preparation	EPFL announced the posibility of transforming its semester project through hackthon, I took advantage of this to do two hacathons, the one in zurich and the one at EPFL. I developed a personal project but I didn't find an alternative in Computer vision for my project.
8	Normal week	Literature review, continue implementation of segmentation (Last semester project) and annotation into stages for object detection (However, the next week we decided to use object detection, and the previous annotation are not useful...)
9	Normal week	Change of supervisior to better suit the project needs. Literature review
10	Normal week	Literature review, choice of methods, research of Dataset
11	Normal week	Annotation, Implementation and report
12	Normal Week	Annotation, Implementation and report
13	Normal week	Implementation and report
14	Normal week	Implementation and report
15	First week revision	Report

Table XIII: Main requirements of the subsystem

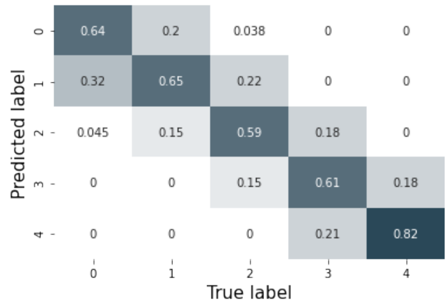
Lessons Learned

As this project is part of an inter-disciplinary project, aiming to promote work between several types of engineers and certain soft skills, this section presents the pedagogical acquisitions.

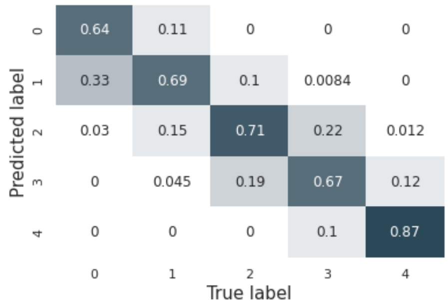
D. Additional figures



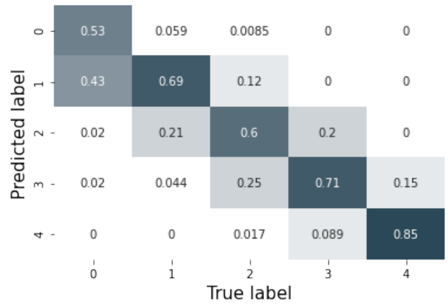
(a) ResNet50 seed 1



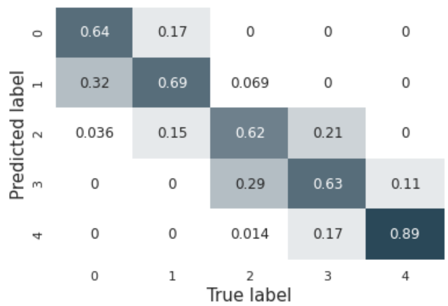
(b) Ordinal regression seed 1



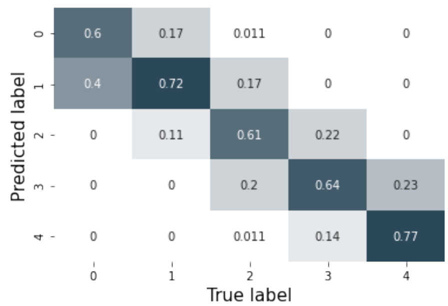
(c) ResNet50 seed 2



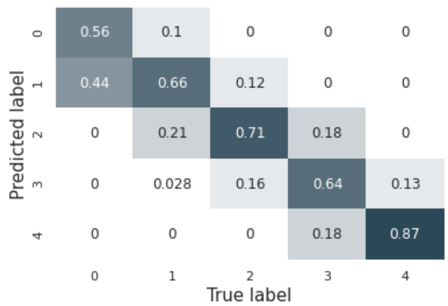
(d) Ordinal regression seed 2



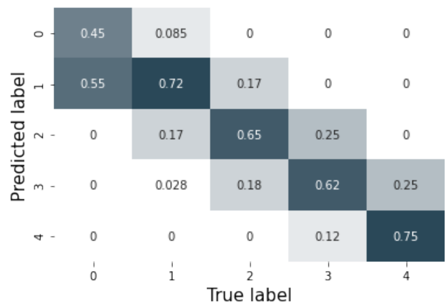
(e) ResNet50 seed 3



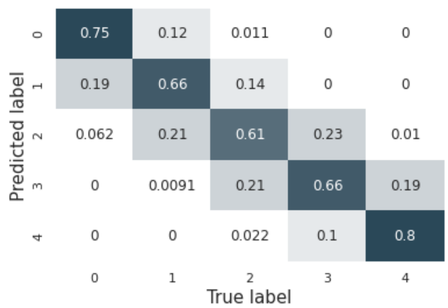
(f) Ordinal regression seed 3



(g) ResNet50 seed 4



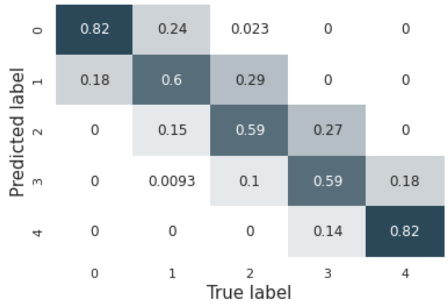
(h) Ordinal regression seed 4



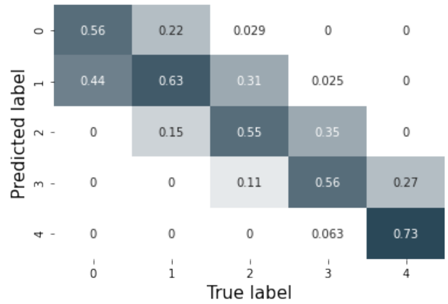
(i) ResNet50 seed 5

(j) Ordinal regression seed 5

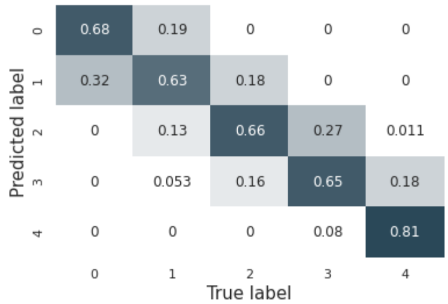
Figure 9: Confusion matrices for a ResNet50 with and without Oridnal regression finetuned on the plant dataset for classification. The data was split randomly 5 times.



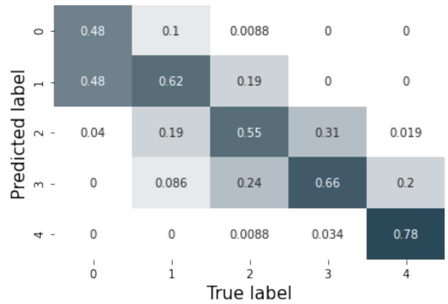
(a) ResNet50 seed 1



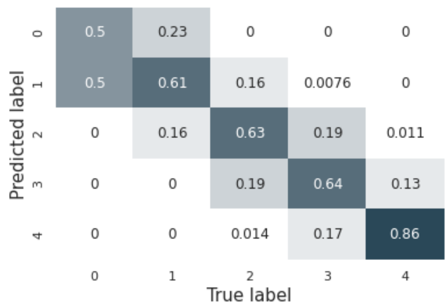
(b) Ordinal regression seed 1



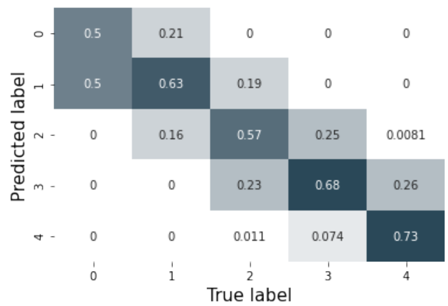
(c) ResNet50 seed 2



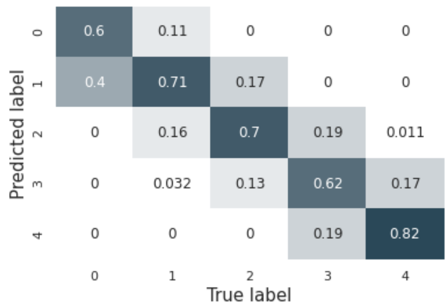
(d) Ordinal regression seed 2



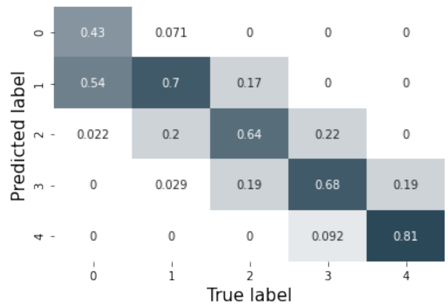
(e) ResNet50 seed 3



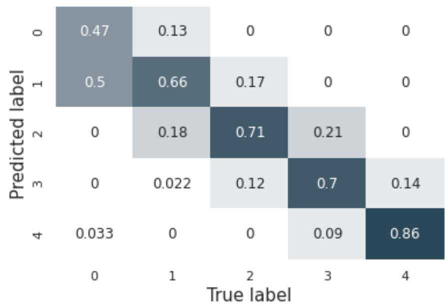
(f) Ordinal regression seed 3



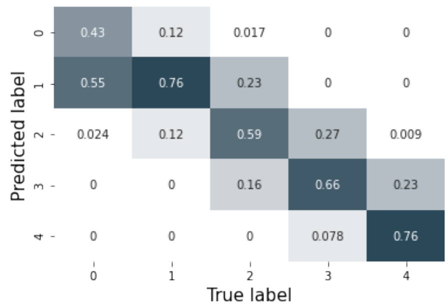
(g) ResNet50 seed 4



(h) Ordinal regression seed 4



(i) ResNet50 seed 5



(j) Ordinal regression seed 5

Figure 10: Confusion matrices for a ResNet50 with and without Oridnal regression finetuned on the plant dataset for classification. The data was split randomly 5 times with cropping.