AI Course

Team Project Final Report

For students (instructor review required)

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background information

Algerian population is expected to reach **70 millions** by 2050 [1], which represents an increase of **65%** compared to 2020. Additionally, urbanization is leading to a drastic decrease in the available farmland. Therefore, this will lead to an increase in food requirements whilst farming spaces will be steadily decreasing. This is why it is important for farmers to find solutions allowing them to optimize the yield of their lands and minimize losses.

In 2016, cereal crops losses due to unpredictable weather conditions were estimated at **USD 2.1 million** [2]. Indeed, the recent climatic changes due to global warming led to an increase in temperature of more than **1.5°C** in Algeria [1], which is more than twice the global temperature increase (estimated at 0.74°C). The unpredictable changes of these climatic conditions make it extremely challenging for farmers to correctly water and fertilize their plants, so much that, in France, **372 farmers committed suicide** in 2015 [3] due to considerable losses engendered by unpredictable weather changes.

Requirements of crops can highly vary from a region of the land to another. However, with traditional farming techniques, crops are globally treated, as if the requirements were uniformly distributed across the land. For example, traditional watering techniques can lead to an over-irrigation of some regions and an under-irrigation of some others. Analogically, farmers spray the entirety of their land with pesticide and other harmful chemical products to treat crop diseases and weeds. And this is because it's hard to tell which part of their land is affected.

Precision agriculture is a technology-enabled, data-driven sustainable farm management system [4]. It is basically the adoption of IoT based devices like smart sensors for data collection, and robots endowed with artificial intelligence for data analysis and autonomous decision-making. The main purpose of precision farming is to maximize the crop yield whilst minimizing the production cost and the environmental effects.

Our idea is to create an intelligent swarm of robots capable of identifying the specific requirements of the different chuncks of the farmland and either treat it locally, or report the needs to the farmer, thus, allowing informed decision making.

1.2 Motivation and Objective

The purpose of our project is to assist farmers and allow them to get the best out of their lands with an efficient use of ressources. We strongly believe that, if concretized, our idea could have a multidimensional impact in different sectors of society :

1.2.1 Economic impact

The main purpose of the entirety of the built-in modules of our system is to maximize the yield of the farmers, whether it is by maximizing the production while recommending the appropriate crop, or by preventing considerable losses while treating the crop diseases and weeds. This would allow a better production, and consequently, and greater economy.

1.2.2 Social impact

people would get in their meals healthier food. Furthermore, if the production increases, food could get more affordable for all social classes.

1.2.3 Ecological impact

With precision farming, pesticides and other chemical product would be applied precisely according to the need instead of spraying all the farmland. Thus, our solution highly contributes to reducing pollution.

1.2.4 Agricultural impact

There already exist sophisticated precision farming tools relying on satellite images and other high-tech materials. These solutions are very expensive and require a lot of technical knowledge. Being autonomous and financially affordable, our solution will radically change the traditional agriculture.

2 Project Execution

2.1 System Diagram

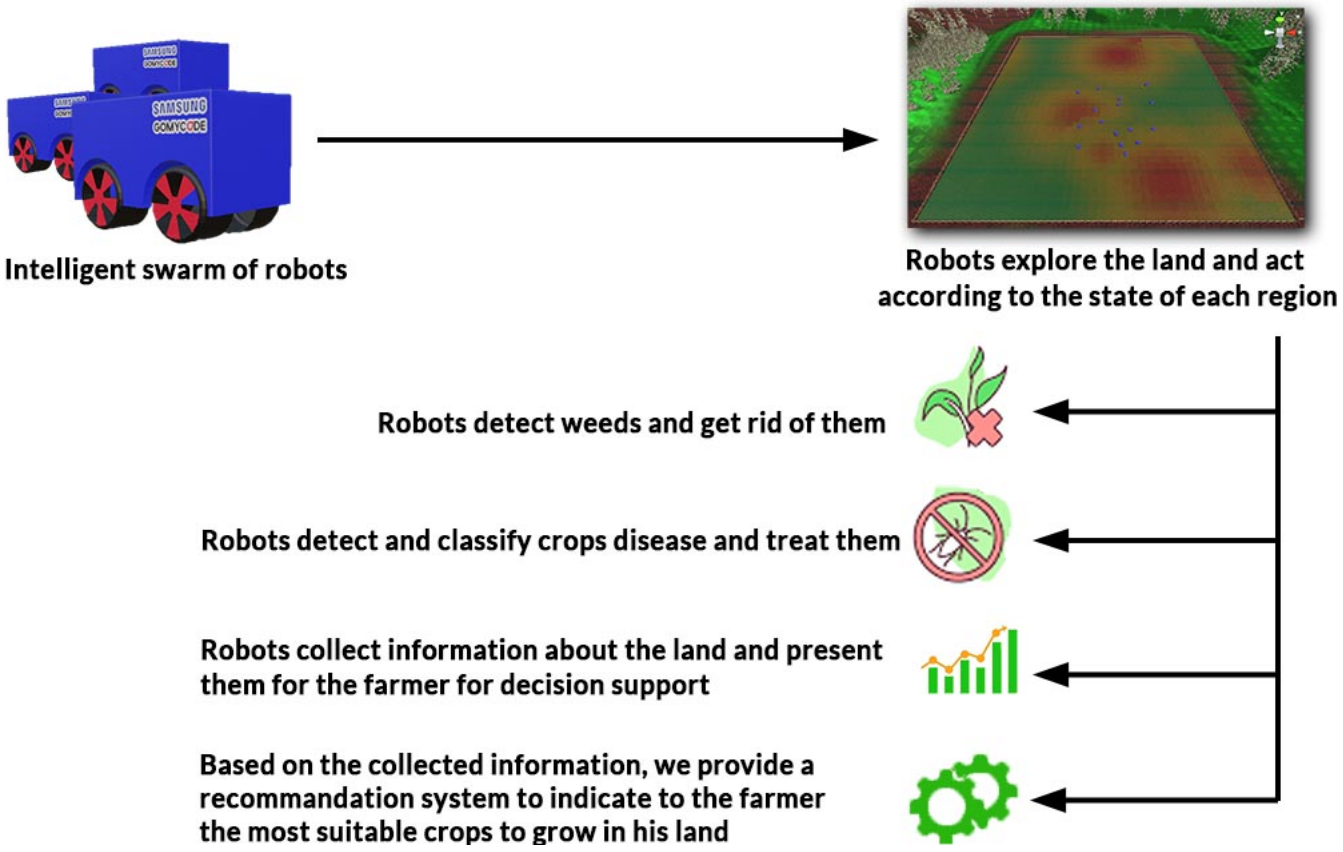

Figure 1: System Diagram

2.2 Swarm intelligence

Swarm Intelligence is the collective behavior of decentralized and self-organized natural or artificial systems. More precisely, it is generally a collective behavior resulting from local interactions between several individuals or with their environment.[5] In our case, we are using robots as a swarm to create collective intelligence to find portions of the land which require the most intervention and make the robots collaborate to treat the problems or report them. Several bio-inspired algorithms have been tried and modified to fit our problem constraints.

2.2.1 Simulation

Simulations are a great tool to prototype a project. Therefore, A 3D simulation has been implemented by the team using Unity game engine to test our algorithms and visualize what would be the final result.

2.2.2 Bio-inspired algorithms

We have implemented two algorithms in our simulation : Bee Swarm Optimization and Particle Swarm Optimization. Of course, we have adapted them to fit our problem constraints, for example :

- Robots can only navigate within the space limited by the boundaries of the farmland .
- Robots need to avoid obstacles.
- Robots have to be able to search another portion after fixing the portion they are working on.

2.2.3 Fitness function

A fitness function is a function that each robot will use to evaluate how well it is performing. Since our goal is to find the portion of the land with the highest amount of needs; the robots would collect data like soil humitidy and PH values or weed presence via the appropriate sensors. Therefore, the fitness function would be a weighted sum of those needs.

$$fitness = \sum weight_i * need_i$$

2.2.4 Particle Swarm Optimization

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming [6]. The algorithm 1 explains how it works.

Algorithme 1 : Particle Swarm Optimization

```
// Best position reached for a given robot.
Array PBest[nb_robots];
// Best position reached among all the robots.
Position GBest;
while True do
    // When the robots are done with region they pass to another
    Reinitialize PBest and GBest;
    while problem not fixed do
        update_GBest();
        for each robot do
            update_PBest();
            velocity = calulate_velocity();
            update_position(robot, velocity);
```

2.2.5 Bee Swarm Optimization

Bee Swarm Optimization algorithm is another bio-inspired algorithm which is inspired from bees strategies to find food in gardens. This algorithm has shown great performance in a lot of use cases. The algorithm 2 explains how it works.

Algorithme 2 : Bee Swarm Optimization

```
Position ref_robot;
while True do
    // Initialize position of the reference robot
    while Problem not fixed do
        // set the robots positions from the ref robots
        set_robots_positions();
        // make the robot with best position as reference bee
        update_positions);
```

2.2.6 Workflow

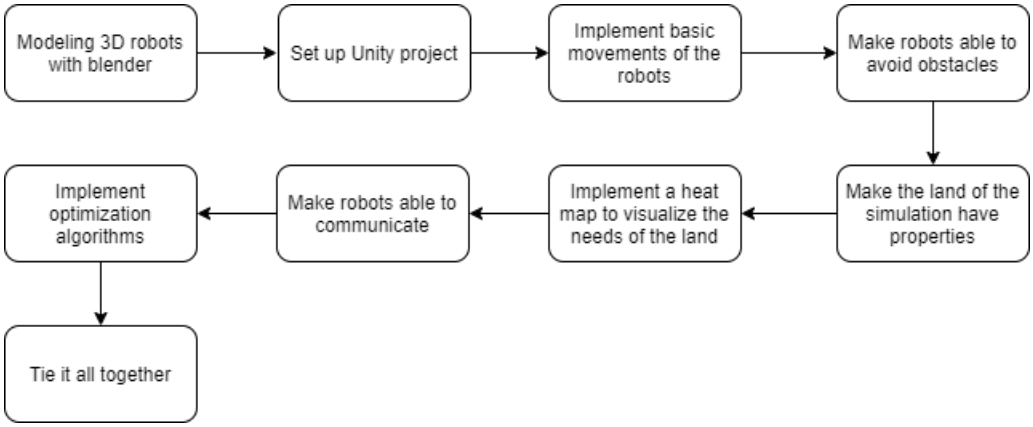

Figure 2: Simulation workflow

2.3 Crop recommendation

The healthy growth of crops highly depends on the value of some parameters such as the macronutrients available in the soil, as well as the ph-value, humidity and the overall temperature and rainfall of the region where the farm is located. However, the appropriate parameters may significantly vary from one crop to another. And therefore, choosing the right crop based on these parameters has proven to be a crucial step in the farming cycle in order to maximize the yield and optimize the product quality. And yet, very often, farmers do not make the right decisions when it comes to choosing the appropriate crop for a specific soil configuration, thus, leading to a serious setback in productivity.

We proposed to address this problem of the farmers through a crop recommendation system to assist them in this important operation. We implemented a machine learning classifier that takes the earlier mentioned soil and environment parameters as inputs, and according to those features, makes a recommendation of the appropriate crop that the farmer should plant.

2.3.1 Data acquisition

For a good and realistic classification, we had to use a dataset that contained as many relevant features for the problem as possible (nutrients, humidity, temperature... *etc*). After going through several datasets, we chose to use the "Crop Recommendation Dataset" from kaggle, which is open source and free to use.

The dataset consists of 2200 instances that contain 7 features and a labeled target which consists of 22 types of crop :

- N, P, K: respectively the ratio of Nitrogen, Phosphorous and Potassium content in soil
- temperature, rainfall: temperature and rainfall of the region respectively in degree Celsius and mm.
- humidity: relative humidity in %
- ph: ph value of the soil
- label: the recommended crop for that specific configuration of features

2.3.2 Exploration Data Analysis (EDA)

The different types of crop :

rice, maize, chickpea, kidneybeans, pigeonpeas, mothbeans, mungbean, blackgram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, coffee.

Features distribution:

_ The average values of the features over all the crops:

- Mean ratio of nitrogen in the soil : 50.55
- Mean ratio of Phosphorous in the soil : 53.36
- Mean ratio of Potassium in the soil : 48.15
- Mean temperature in Celsius : 25.62
- Mean Relative Humidity in % is : 71.48
- Mean ph value of the soil : 6.47
- Mean Rain fall in mm : 103.46

_ A visualization of the distribution for agricultural conditions :

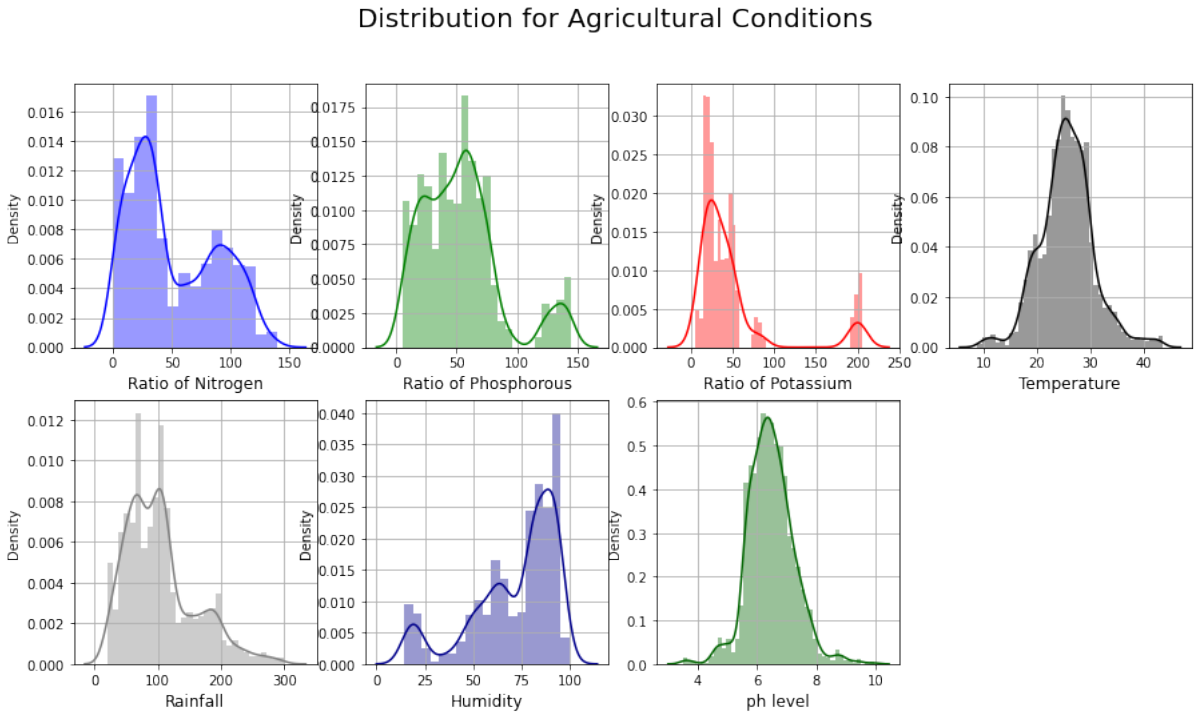

Figure 3: Distribution of features over all the types of crop

From the graphic above, we can observe for example that most of crops require a ph balancing between 6 and 7, and a temperature between 20°C and 30°C.
_ A visualization of the distribution of nitrogen over all the types of crop :

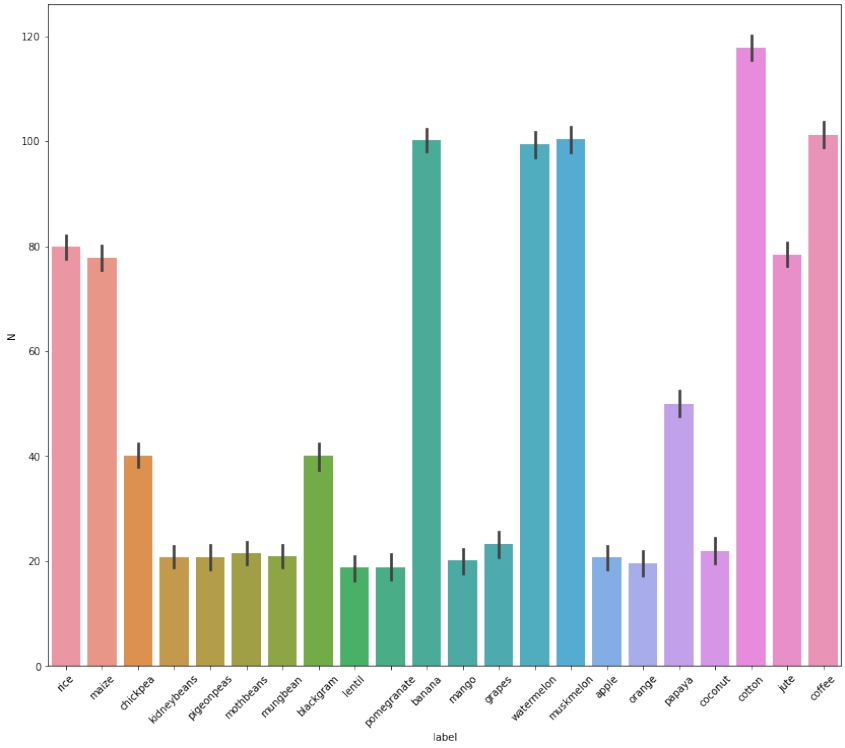

Figure 4: Distribution of nitrogen ratio over all the types of crop

We can clearly observe the variation of needs in terms of nitrogen from one crop to another. For example, banana requires a ratio higher than 95%, whereas lenti only requires 19%.

2.3.3 Modeling

To achieve this recommendation system, we ran our dataset through multiple machine learning algorithms in order to compare their performance and achieve the best possible accuracy scores. Below, you can find a description of the implmented algorithms:

Logistic Regression: Logistic regression is one of the most popular machine learning algorithms, which comes under the supervised learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

K Nearest Neighbors: KNN is a supervised machine learning algorithm, it basically tries to predict the class for the test data based on the classes of the K nearest training data points.

Random Forest: Random Forest is also a supervised machine learning algorithm, built of multiple decision trees that are merged together via the bagging method,in order to get a more accurate and stable prediction.

2.3.4 Results

After running our dataset through the above mentioned machine learning algorithms, we obtained the following results:

Algorithm	Logistic Regression	K Nearest Neighbors	Random Forest
Accuracy Score (%)	96.73	97.82	99.82

Table 1: Different accuracy scores of the implemented algorithms

We can see that all the models performed very well. However, the Random Forest algorithm has a slightly better accuracy than the other two.

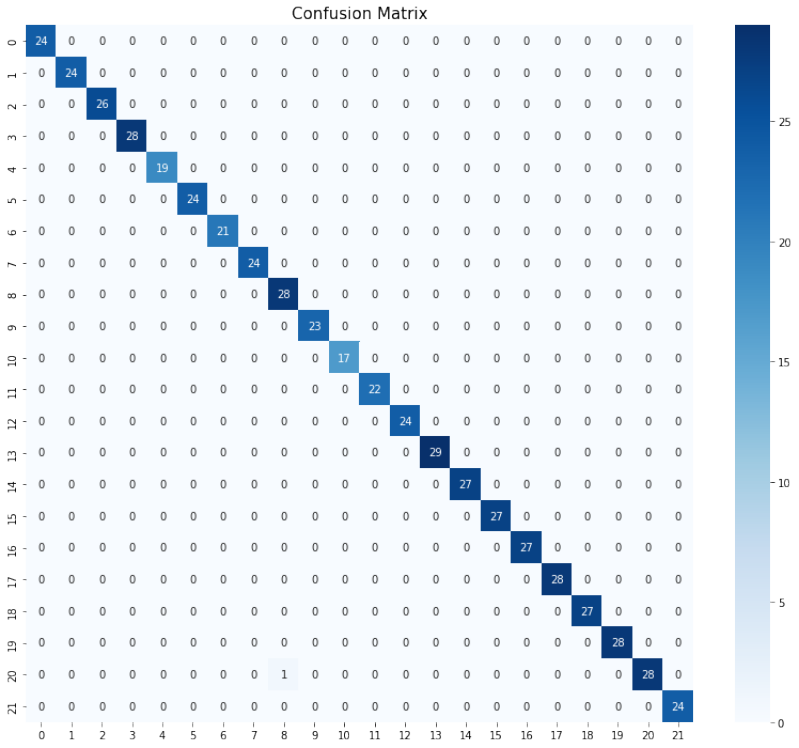

Figure 5: Confusion matrix of the Random Forest model

Conclusion: To sum up, this module allows farmers to make better decisions on which crops should be planted depending on some important parameters. Based on the dataset we used, we have been able to implement a model capable of classifying 22 types of crop. However, in practice, the information we need to make the classification should be collected from sensors and treated before being fed to the model. Therefore, it should be easy to add any type of crop to our model.

2.4 Crop disease detection

Crops and livestock pests, diseases and infestations are a major stress for the agricultural sector. According to the estimation of the Food and Agriculture Organization of the United Nations (FAO), “such biological disasters caused 9 percent of all crop and livestock production loss in the period from 2008 and 2018”.

Indeed, when not diagnosed and treated early enough, crop diseases can cause irreversible damages to plants, which is even more alarming as they tend to propagate and infect more and more crops across the land very quickly.

To prevent their crop plantations from being infected by diseases and infested by pests, farmers are, at present, using a lot of pesticides. However, the latter have been shown to be not without danger from human health. So is

it feasible for a farmer to do without these chemicals?

Considering that farmers usually have large fields, each of them with a different sort of crop, which has its own sort of diseases, If we wanted to detect the diseases in a traditional way, we would have to go through each land plant by plant, examine it, and for the process to be useful, do this on a daily basis. Considering also that farmers are not experts in phytopathology (science that studies plant diseases), even with this much effort, their diagnosis might be wrong. Recruiting an expert for this tedious and repetitive task is also not a solution as it would pose an additional charge on the farmer and be a waste of time for the expert.

Our solution is to implement each of our robots with a camera that will regularly collect images of the plants as the robot is browsing the field, and, using computer vision and deep learning, judge whether these plants are infected by any disease, and if so, what disease it is.This would, eventually, make farmer's lives easier, prevent them from an economic loss and contribute to improving food quality as pesticides and treatments won't be used as intensively as before and applied only locally in infected zones at the very early stages of the disease and not all over the field.

As a proof of concept, we are going to build a disease detection model for one crop type (Tomato) diseases only. This can then be generalized to other crop species.

2.4.1 Data acquisition

There are currently two public plant diseases datasets available : PlantVillage Dataset (PVD) and Plantdoc Dataset (PD).

PlantVillage Dataset

PlantVillage is the oldest and most common public dataset for plant disease classification; it consists of 54,303 healthy and unhealthy leaf images divided into 38 categories by species and disease. It has been used by its creators to build an automated system using GoogleNet and AlexNet for disease detection, achieving an accuracy of 99.35 percent. However, the images in PlantVillage dataset are taken in laboratory setups and not in the real conditions of cultivation fields, therefore its efficiency in the real world is likely to be poor.

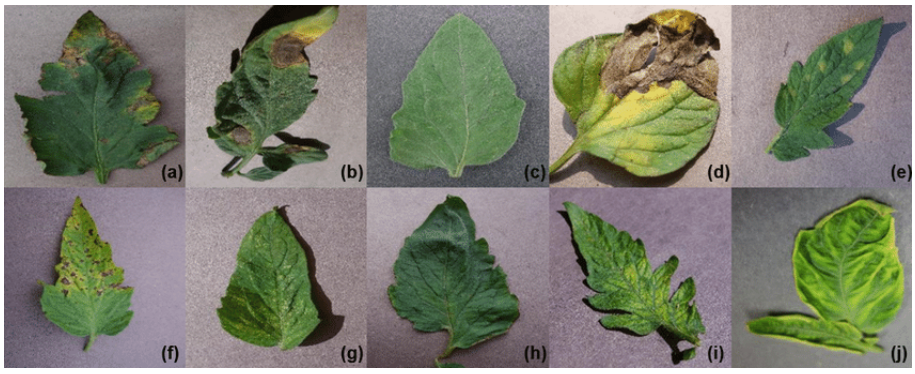

Figure 6: Sample from PlantVillage Dataset

The dataset is available on Kaggle.

Plantdoc Dataset

Plantdoc Dataset has been created recently (2020). It contains 2,598 data points in total across 13 plant species and up to 17 classes of diseases, which makes 2.5 times less data than PVD, but under a more realistic form, since images are not taken under controlled settings. The dataset was published by its authors on github :

Datasets	Advantages	Disadvantages
PlantVillage Dataset	- More Data - Community support (many projects have used it) - Getting a high accuracy easily	- Lab-controlled images, which limits the effectiveness of detecting diseases in field in real-time
Plantdoc Dataset	- Images in the real conditions of cultivation fields, which makes it suitable for training our robots	- Not enough data - Complex - Few projects have been made using this dataset - Hard to get a good accuracy

Table 2: Comparison between PlantVillage and Plantdoc datasets

- PlantDoc-Dataset for leave disease classification (without annotations of bounding boxes to delimit the leaves).
- PlantDoc-Object-Detection-Dataset for plant leave detection (each image is accompanied by an annotation XML file which gives coordinates of upper-left and downer-right corners of the rectangular bounding boxes that delimit the leaves in the image).

Figure 7: Sample from Plantdoc Dataset

In the below table (Table 2), we draw a comparison between the two datasets.

PlantifyDr Dataset : In addition to PlantVillage and Plantdoc datasets, there is a dataset on Kaggle that combines images from both of the previous datasets for a total of 125,000 leaf images augmented and distributed over 10

different plant species. The dataset is organized for each species in 'train' and 'valid' folders. Train Folder contains an unbalanced mix of images from PlantVillage and Plantdoc (majoritarily PVD) and Validation Folder comprises images from Plantdoc only (figure See PlantifyDr Dataset on Kaggle.

- Training data : As for the training data, we tried using only images in complex background (Plantdoc) or a combination of plain and complex background images (PlantifyDr dataset)
- Testing data : Since the images the robots are going to take will be photos in real cultivation conditions with complex background, it appeared to us necessary that the testing data for our model should be taken from Plantdoc dataset. However, for the sake of comparison, we did use a testing set that contained a majority of lab-controlled images too

Testing Set Images \ Training Set Images	controlled conditions plain background + some real images	real images complex background
controlled conditions plain background + some real images	We will do it for the sake of comparison	pointless
real images complex background	Generalization is difficult, but data is available	Ideal, but requiring a lot of data which we do not have

Table 3: Approaches in choosing training and testing data

2.4.2 Training methodology

Classification VS Detection

Let us first of all clarify the difference between the notions of image classification and object detection. Classification basically answers the quetsion *"**What** is the class of the object present in the image?* Detection on the other hand, searches for the class and location of the instances of any particular object in the image *"**Where** are the objects in the image and **what** are their respective classes?"* The detection takes, basically, the classification a step further by adding localisation of the objects of interest

For our project, the images that the robot will take are going to be quite different from one another, they will vary in : number of leaves in the picture, light, colour, size of leaves, background, etc. All these variations could be (mis)interpreted as features by a convolutional neural network trained for a Classification task, and thus it is quite difficult to make such a model perform well

On the other hand, in Plant Leaf Disease Detection, variation in number of leaves won't affect the model as the latter will detect the leaves and classify them individually. Other sources of variation (light, size, orientation,etc) can be treated by data augmentation. The problem is that building and training such a model is computationally intensive, time consuming, and a great amount of annotated 'real-life' data is needed to make it performant

Training methodology for classification

Three parameters were to be considered in choosing and establishing our training methodology :

- Architecture of the model : we opted for Convolutional Neural Networks since we are dealing with images only as inputs
- Whether to use Transfer Learning or not : It is preferable to use Transfer Learning since we lack data and computational resources. We used a pretrained MobileNet model as a backbone to our model
- What data to use : We summarize the three main approaches in Table 3.

Training methodology for object detection

Among state-of-the-art methods for object detection, there is Faster Region-based Convolutional Neural Network (aka Faster R-CNN), which we chose to implement in this project.

"In this method, the detection process is carried out in two stages. In the first stage, a Region Proposal Network (RPN) takes an image as input and processes it by a feature extractor. Features at an intermediate level are used to predict object proposals, each with a score. For training the RPNs, the system considers anchors containing an object or not, based on the Intersection-over-Union (IoU) between the object proposals and the ground-truth. In the second stage, the box proposals previously generated are used to crop features from the same feature map. Those cropped features are consequently fed into the remaining layers of the feature extractor in order to predict the class probability and bounding box for each region proposal. The entire process happens on a single unified network, which allows the system to share full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals." [7]

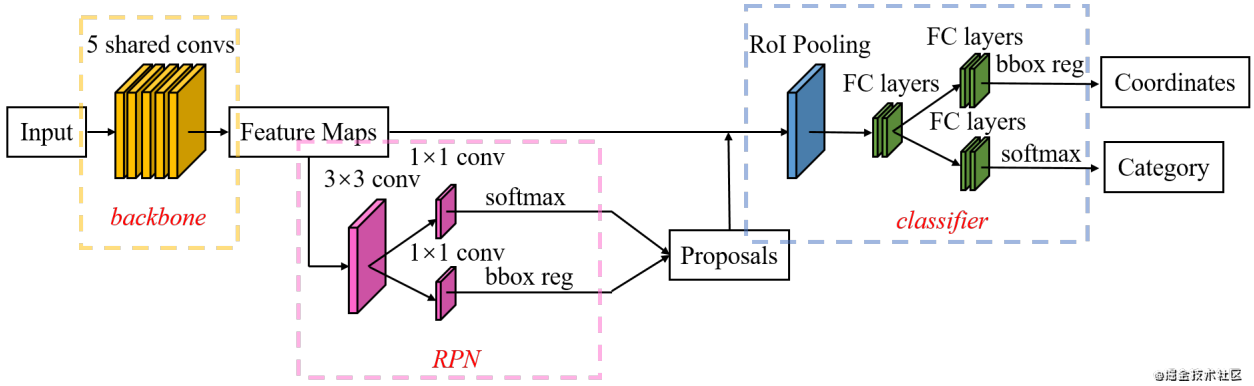

Figure 8: Faster-RCNN architecture[1]

2.4.3 Exploratory Data Analysis (EDA) and Data Preprocessing

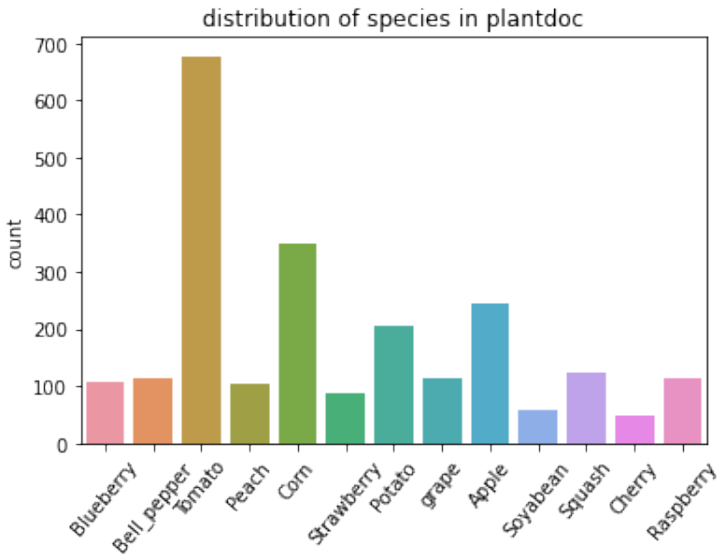

Figure 9: Distribution of crop species in Plantdoc Traing Set

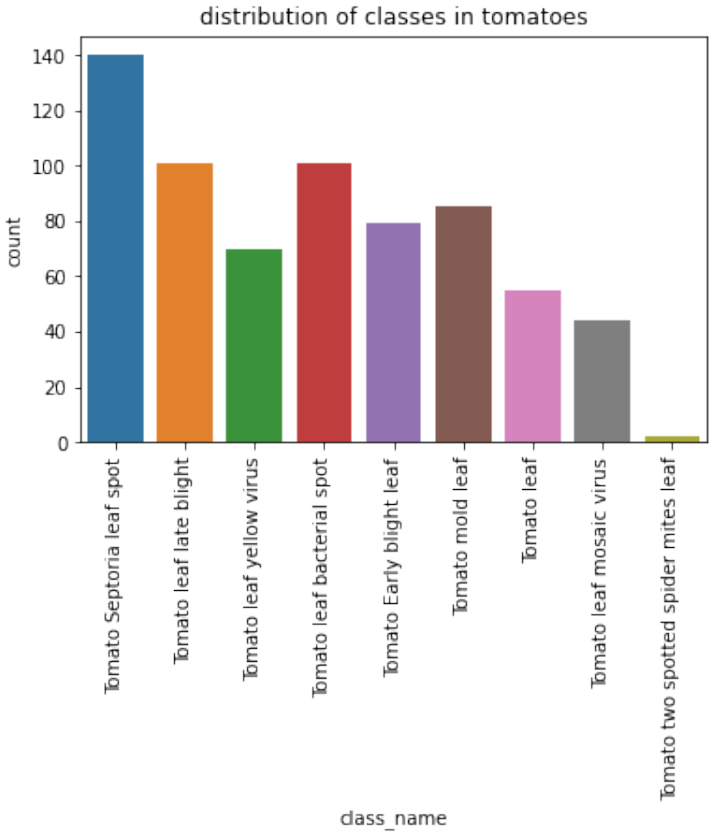

Figure 10: Distribution of Tomato diseases in Plantdoc Traing Set

Datasets for classification task : Plantdoc and PlantifyDr

Figure 10 shows that there are less than 700 images of tomato inside the Plantdoc dataset, distributed as shown in Figure 11.

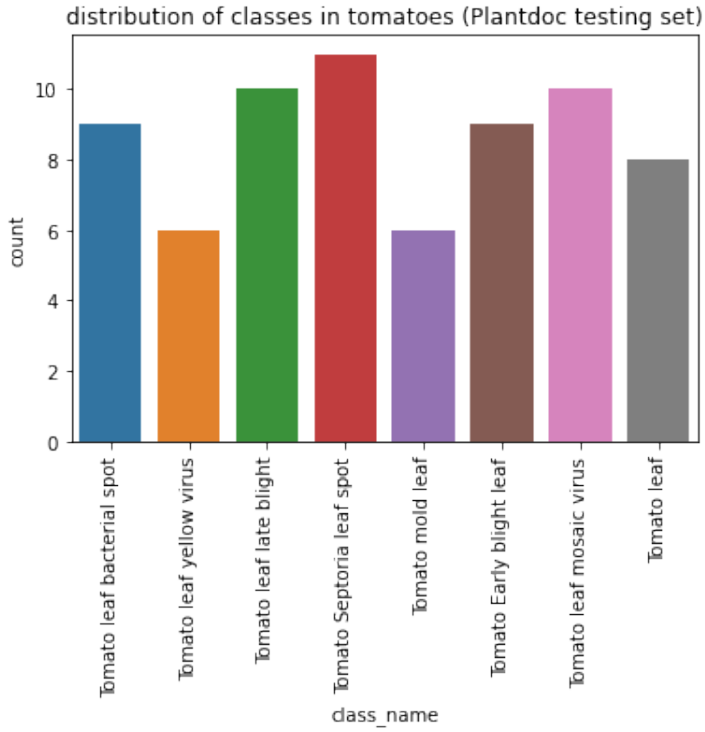

Figure 11: Distribution of Tomato diseases in Plantdoc Testing Set

Figures 11 and 12 show that there are only two images of the category "Tomato two spotted spider mites leaf" in the whole dataset, it then should be removed

Figure 12: Distribution of Tomato diseases in PlantifyDr Training Set

We can notice from Figure 13 that classes in PlantifyDr Training Set are balanced and rich in images on the whole

Data for object detection and its preprocessing

We created a training dataset for Faster-RCNN from manually mixing tomato images from Plantdoc-Object-Detection dataset and from PlantVillage dataset and then unifying the classes names. To be then inputted to our Faster R-CNN network, the data needed to be formatted in text files where lines had the following structure: (filename,xmin,ymin,xmax,ymax,class_name) where

- Filename : path of the image location
- xmin,ymin,xmax,ymax : coordinates of the top-left and bottom-right points in the bounding box
- class_name : class of the object inside the bounding box

Note : We could not use use PlantifyDr Dataset because images it contained were not annotated (it was designed for classification task only and not detection). As for the PlantVillage images annotation, since images only contained one leaf per image, we added bounding boxes by taking a margin of 20 px from the image border.

2.4.4 Modeling

Classification of Tomato plant diseases

1. **Training and testing using Plantdoc (realistic images only)**
2. **Training with majoritarily lab-controlled images and testing on realistic images**

   In both previous cases :

   - **Model Architecture:**
     We used a MobileNet as a backbone to our model, to which we added two fully connected layers along with two Dropout layers to reduce overfitting
   - **Model configuration:**
     We used Adam optimizer with a learning rate of 0.001
3. **Training and testing on set with a majority lab-controlled images**

   - **Model Architecture:**
     We used a MobileNet network trained on PlantVillage dataset as a frozen backbone to our model (feature extractor) to which we added two fully connected layers along with two Dropout layers to reduce overfitting The pretrained model was obtained from a notebook on Kaggle [2].
   - **Model configuration:**
     We used Adam optimizer with a learning rate of 0.001

Detection of Tomato plant diseases We recreated the architecture described in section 2.4.2 from scratch using Keras.

[2] https://www.kaggle.com/thunder2901/leaf-disease-classification-mobilenet/data

2.4.5 Results

Classification of Tomato plant diseases

We summarize results obtained from each of the following approaches in Table 4.

1. **Case 1** : Using Plantdoc for training and validation
   In this case, we trained and tested our model on realistic images only. Results are shown in Table 4.
2. **Case 3** Using both train and valid sets in PlantifyDr-Dataset-Tomato for training and validation
3. **Case 2** Only using train set in PlantifyDr dataset

Choosing of training and validation sets	Accuracy
Case 1	0.46
Case 2	0.45
Case 3	0.92

Table 4: Evaluation of classification models

Figure 13: Correct classification

Figure 14: Incorrect classification

Detection of Tomato plant diseases

Our model performed poorly because of the lack of realistic images taken under uncontrolled conditions.

2.4.6 Encountered problems and further improvements

"In real complex natural environment, plant diseases detection is faced with many challenges, such as small difference between the lesion area and the background, low contrast, large variations in the scale of the lesion area and various types, and a lot of noise in the lesion image."[8] The only way to tackle these difficulties when aiming to build a performant detection model is by providing it with a big amount of labeled, well annotated images taken in real complex environment. This also implies that more powerful computational resources are needed to process all of that data in a reasonable amount of time.
Indeed, Fuentes et al.[7], for example, got an accuracy of 83 % when collecting their own complex data (5000 images of tomato leaves in complex environment) and training their model on it.

2.5 Weeds detection

Weeds are plants that grow in undesirable places, and the reason why they are undesirable is that they often compete and win against crops for space, light, water, and nutrients, reducing harvest efficiency. Traditional management strategies deployed by farmers to reduce the impact of weeds can damage the crops and harm the environment by using some dangerous chemicals in an improper way.

Weeds annually cause yield **losses of 2.7 million tonnes** [9] and the cost for weed control service is between **$65** and **$150** per treatment. In Algeria more than 40 variety of pesticides are widely used by farmers. They use 6000 to 10000 T/year [10] of pesticides for an estimated cost of **4.5 billion dinars** annually, which makes Algeria a large consumer of pesticides. So, implementing a 'weed detection from crop ' model can be beneficial to improve the product quality and minimize agrochemical residues in our food.

2.5.1 Data acquisition

The dataset we used is an open source dataset **"V2 Plant Seedlings Dataset"** available on **"The Computer Vision and Signal Processing Group, Department of Engineering – Aarhus University"** website. This dataset contains **5,539** good quality images of crop and weed seedlings, each image contains different plants with a realistic and complex background. The images are grouped into 12 classes.
5 classes of crops:
Common wheat, Maize,Shepherd's Purse, Small-flowered Cranesbill, Sugar beet.
7 classes of weeds:
Black-grass, Charlock, Cleavers, Common Chickweed, Fat Hen, Loose Silky-bent, Scentless Mayweed.

2.5.2 Exploration Data Analysis (EDA)

some pictures from the dataset with their labels

Figure 15: Data With Label

Correlation Matrix

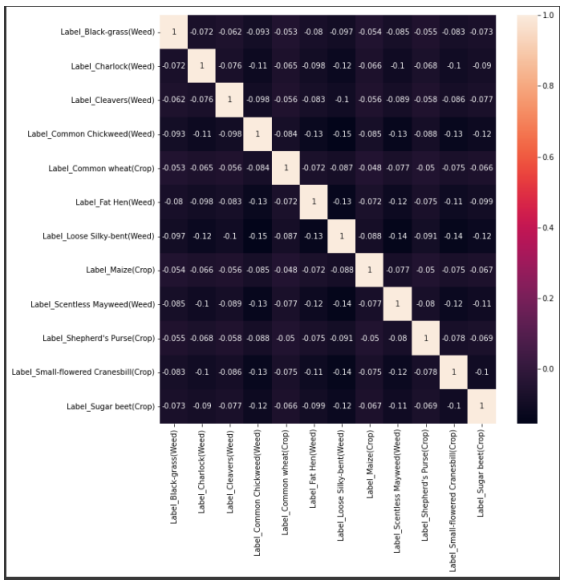

Figure 16: Correlation Matrix

data distribution over classes

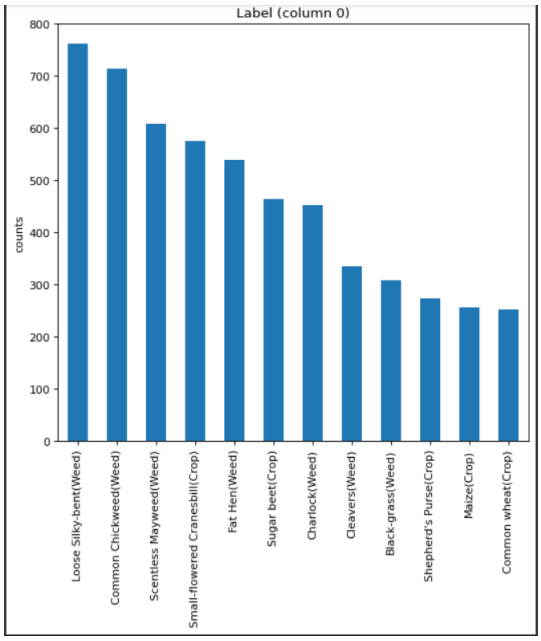

Figure 17: Data Distribution

2.5.3 Training method

Our data being a set of pictures we decided that we would use Convolutional Neural Networks . After reading some papers we decided that we would use transfer learning with MobileNetV2.
MobileNetV2 is a CNN architecture model for image classification that was originally trained on the ImageNet dataset(ILSVRC2012).it is very similar to the original model MobileNet except that it uses inverted residual blocks with bottlenecking features.it support any input size greater than 32 x 32, with larger image sizes offering better

performance.

There are other existing models that we could have used but what made us choose MobileNetV2 specially is that it has a very less computation power. This model requires 9 times less work than similar neural networks while achieving a similar accuracy.

2.5.4 Modeling

MobileNetV2 Architechture:

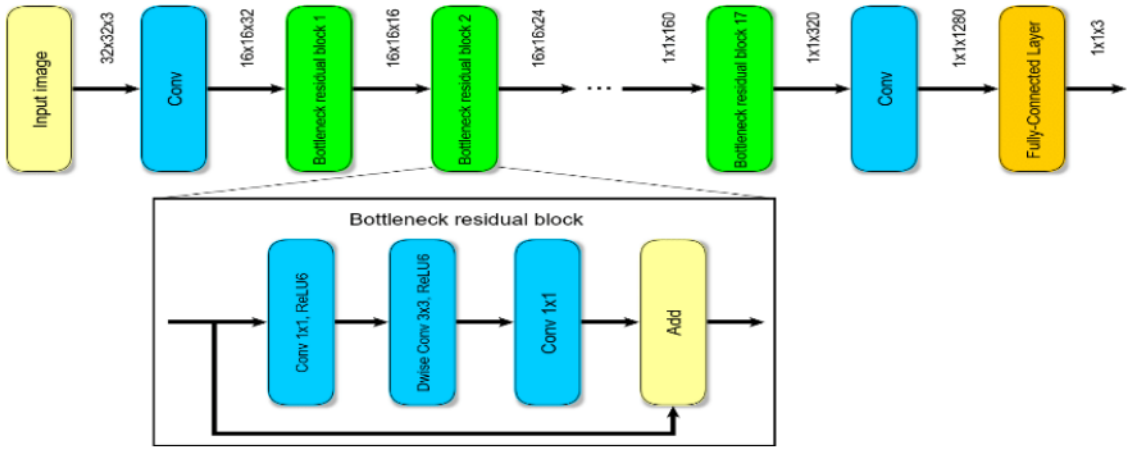

Figure 18: MobileNetV2 Architechture

Additional layers:
We did add 3 fully-connected layers, each neuron will be connected to all the neurons of the previous layer and each of the nodes of the fully-connected layers will output a score corresponding to a class score.
Our last dense layer has an output shape of 12 so we would get 12 output scores since we have 12 classes in our DataSet

2.5.5 Data Preprocessing

Since this dataset has been regularly used in competitions, we didn't need to make any changes due to its initial performance.
we started by creating a pandas dataframe with 2 columns: image_path and image_label then we create a train generator and a test generator using the keras preprocessing imageDataGeneration function.
and by applying this two generators to our data frame we obtain our training, testing and validation image sets with real time data augmentation

2.5.6 Testings and improvements

The obtained results after training our model are shown in Figure 19 and 20 below

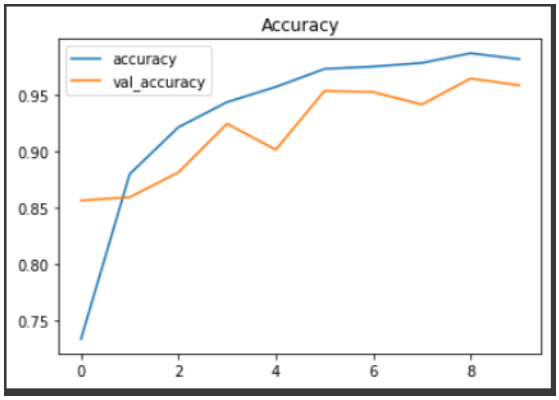

Figure 19: Accuracy

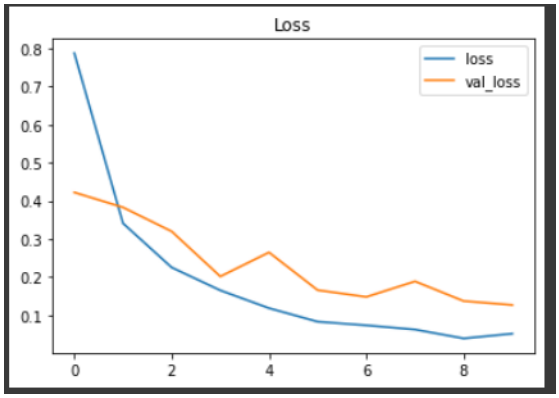

Figure 20: Loss

We can see that the accuracy is getting better over epochs which shows that our model is not underfiting nor overfiting. For the test our model got an **Accuracy** score of **0,9585** and a **Loss** of **0,14383**.

Classification Metrics:

	precision	recall	f1-score	support
Black-grass	0.91	0.84	0.87	57
Charlock	0.99	0.97	0.98	103
Cleavers	1.00	0.98	0.99	61
Common Chickweed	0.97	0.99	0.98	161
Common wheat	0.97	0.94	0.95	65
Fat Hen	0.99	0.98	0.98	97
Loose Silky-bent	0.94	0.98	0.96	140
Maize	1.00	1.00	1.00	55
Scentless Mayweed	0.87	0.98	0.92	121
ShepherdтAllls Purse	1.00	0.72	0.84	57
Small-flowered Cranesbill	0.95	0.99	0.97	109
Sugar beet	1.00	0.98	0.99	82
accuracy			0.96	1108
macro avg	0.97	0.95	0.95	1108
weighted avg	0.96	0.96	0.96	1108

Figure 21: Classification Metrics

Extension :Real Time Classification

We did write a python code that allows us to do real time testing using an external webcam.So that we could take pictures and have a real time classification.

Bonus Tests:

We also wanted to test our model in a more realistic way, in order to do that we went to a nursery and took 30 photos from differents plants types.Among these plants we had some new plant types. In this Hard Test set our model scorecced an **accuracy of = 0,7**

2.5.7 Workflow

Model Workflow

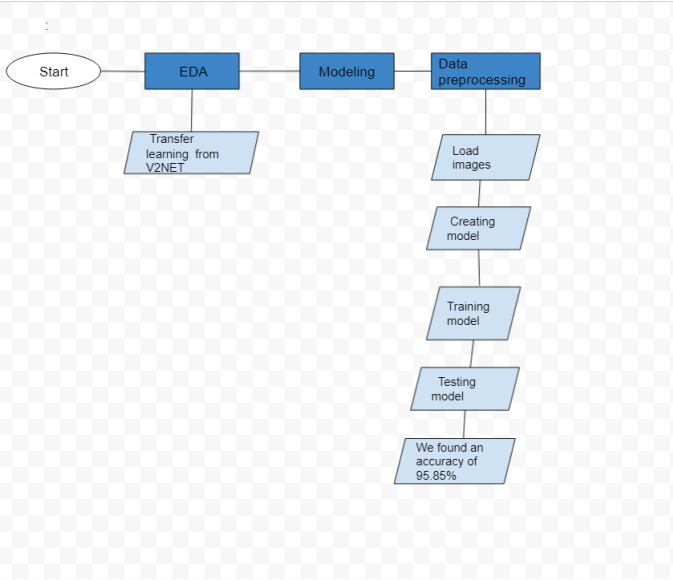

Figure 22: Weed Detection Workflow

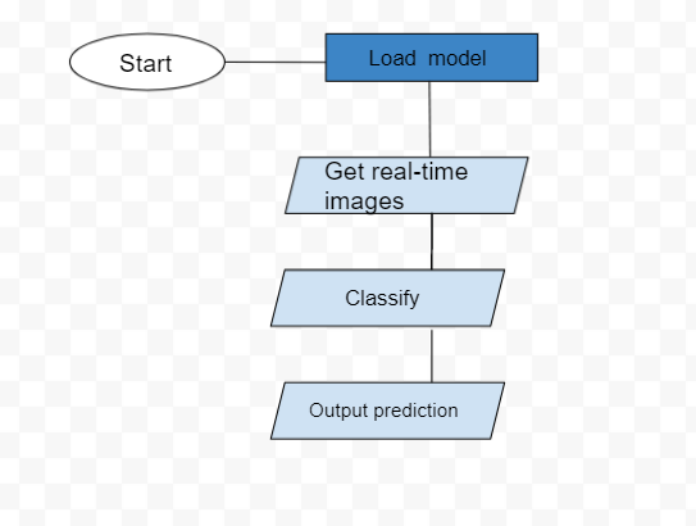

Figure 23: Real Time classification Workflow

3 Projected Impact

3.1 Accomplishments and Benefits

In conclusion, we have been able to implement several modules of precision agriculture that will help farmers improve their yield both in terms of quantity and quality. These features are summarized below :

- Robots navigation using a swarm intelligence algorithm that takes into account the various requirements in different regions of the land and makes the robots work in collaboration in order to optimize their productivity.
- Crop recommendation, which consists of analyzing the soil and environment properties, and then making a recommendation of crop based on a machine learning classifier. The aim of this feature being to garantee the healthy growth of the crops which will lead to an optimal yield.
- Weed detection, which allows the early spotting and treatment of the weeds using computer vision to do the classification crop/weed, which can prevent considerable losses for farmers as these bad plants are usually hard to notice and are extremely contagious.
- Crop disease detection, which is a pretty similar module compared to weed detection, we used computer vision to detect crop diseases and classify them. This will allow an early treatment and a considerable loss prevention.

3.2 Future Improvements

The different modules of our system need to be trained with real data collected from actual sensors to potentially discover issues and fix them. Therefore, with more time and equipments, this project could be implemented on a bigger scale, and eventually, lead to better results.

Although our system gathered several useful modules for farmers that are performing pretty well, there is still room for huge improvements. Here are the improvements we have in mind :

- Connecting our system to the automated irrigation system in order to allow smart irrigation based on the map of requirements generated by the robots' measurements.
- Expanding our system's features to assist the farmer in all the parts of its work by adding a smart harvesting module to reduce the labour, time and cost of harvesting, as well as a smart livestock management to monitor the health and well-being of cattle.
- Implementing obstacle avoidance for the robots using artificial intelligence techniques like fuzzy logic.

References

[1] O. Bessaoud, J.-P. Pellissier, J.-P. Rolland, W. Khechimi. Rapport de synthèse sur l'agriculture en Algérie. [Rapport de recherche] CIHEAM-IAMM. 2019, pp.82. ffhal-02137632

[2] Atlas Magazine: "Algérie : Des pertes agricoles estimées à 2,1 millions USD en 2016",[online] https://www.atlas-mag.net/article/des-pertes-agricoles-estimees-a-230-millions-dzd-en-2016, consulted on 25/09/2021.

[3] Vie Publique: "Suicides dans le monde agricole : comment mieux aider les agriculteurs en difficulté ?",[online] https://www.vie-publique.fr/en-bref/277663-suicides-dans-le-monde-agricole-aider-les-agriculteurs-en-difficulte, consulted on 25/09/2021.

[4] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, "Machine learning applications for precision agriculture: A comprehensive review," *IEEE Access*, vol. 9, pp. 4843–4873, 2021.

[5] Swarm Intelligence : qu'est-ce que l'intelligence distribuée ? [online] https://www.lebigdata.fr/swarm-intelligence-distribuee-definition, consulted on 25/09/2021

[6] KENNEDY, James et EBERHART, Russell. Particle swarm optimization. In : Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995. p. 1942-1948.

[7] Fuentes, Alvaro, Sook Yoon, Sang Cheol Kim and Dong Sun Park. "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition." Sensors (Basel, Switzerland) 17 (2017): n. pag.

[8] Liu, J., Wang, X. Plant diseases and pests detection based on deep learning: a review. Plant Methods 17, 22 (2021). https://doi.org/10.1186/s13007-021-00722-9

[9] frontiers in Agronomy: "Grand Challenges in Weed Management",[online] https://www.frontiersin.org/articles/10.3389/fagro.2019.00003/full, consulted on 27/09/2021.

[10] K. Ait Mohamed, S. Imadouchene, "Contribution à l'étude de l'utilisation des pesticides dans les régions de Fréha et d'Azeffoun (Tizi-Ouzou)", 2017.