Intelligent Small-Scale Strawberry Irrigation System for Different Weather Conditions

(Defense Seminar)

Ye Htet (ME.EcE-4)

Master of Engineering

Faculty of Electronic Engineering

Supervisor

Dr. Htin Kyaw Oo

External Examiner

Dr. Tin Tin Hla

Co-Supervisor

Dr. Theint Theint Soe

Member

Daw Hnin Pwint Phyu

September 18, 2019

Introduction

- Water resources become insufficient in central region of Myanmar.
- To overcome these problems, technology is one of the best solutions.
- People in agriculture need agricultural knowledge to make precise decisions.
- Efficient water and fertilizers usage for strawberry plants to bear fruits all the year round in Pyin Oo Lwin.

Research Questions

- Why strawberry plants can’t bear and produce fruits in all seasons?

- How to grow strawberry plants in different regions?

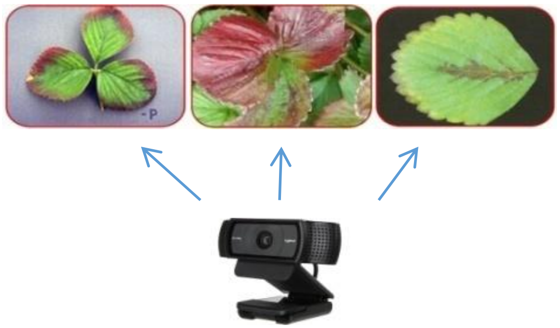
precise temperature, water and fertilizer management

Research Methods

Using modern technology to bear fruits all the year round.

Classifies plant leaves by using image processing.

Using automatic drip irrigation system to water and feed fertilizers.



Outline of Presentation

- Introduction
- Research Questions
- Research Methods
- Overall Block Diagrams
- Small-scale Farm Structure and Design
- Nutrient Deficiency Symptoms Detection System ☆
- Drip Irrigation System ☆
- Temperature Control System ☆
- Discussion
- Conclusion

System Overview Design

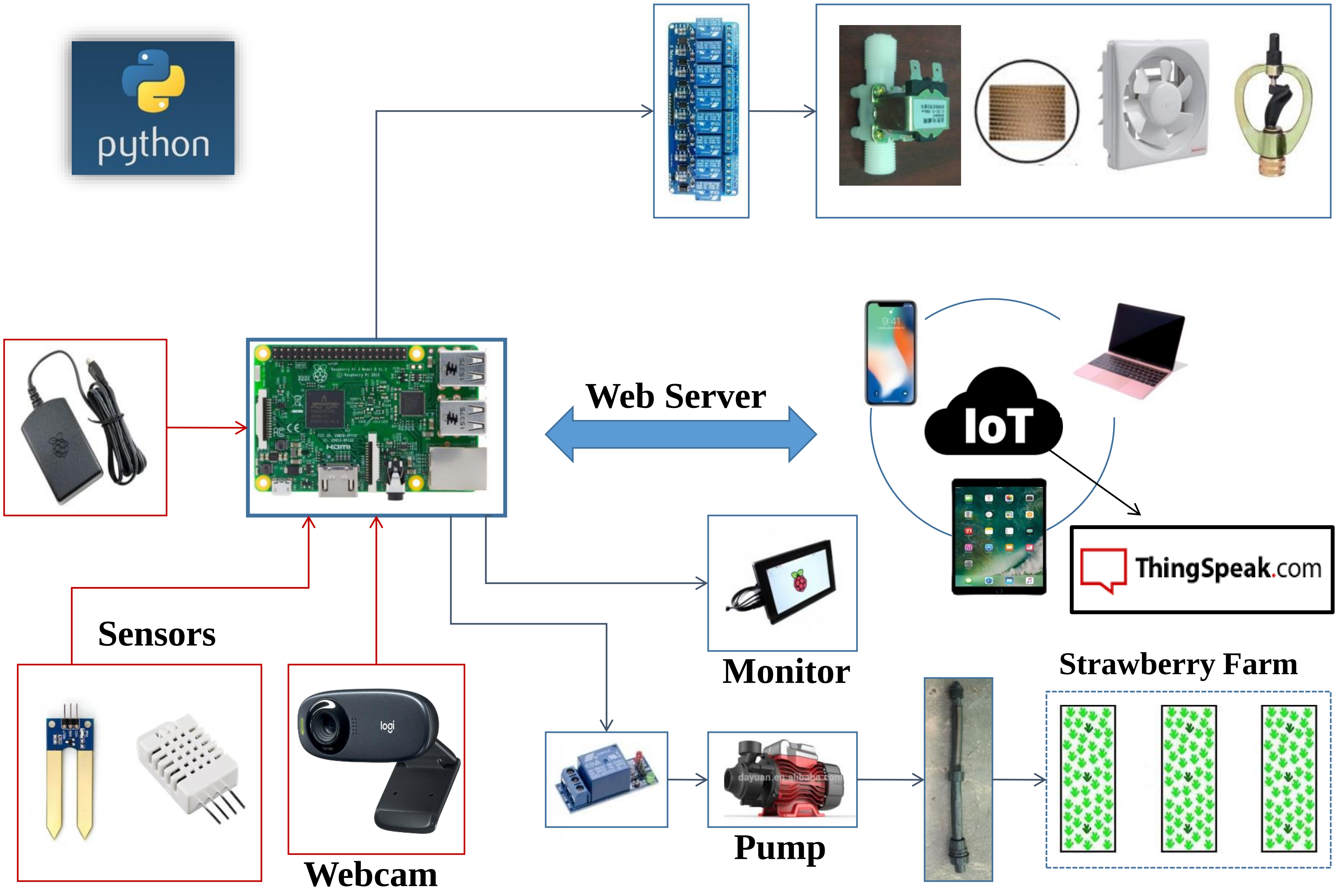


Image Processing Steps

Nutrient deficiency symptoms detection (NPK)

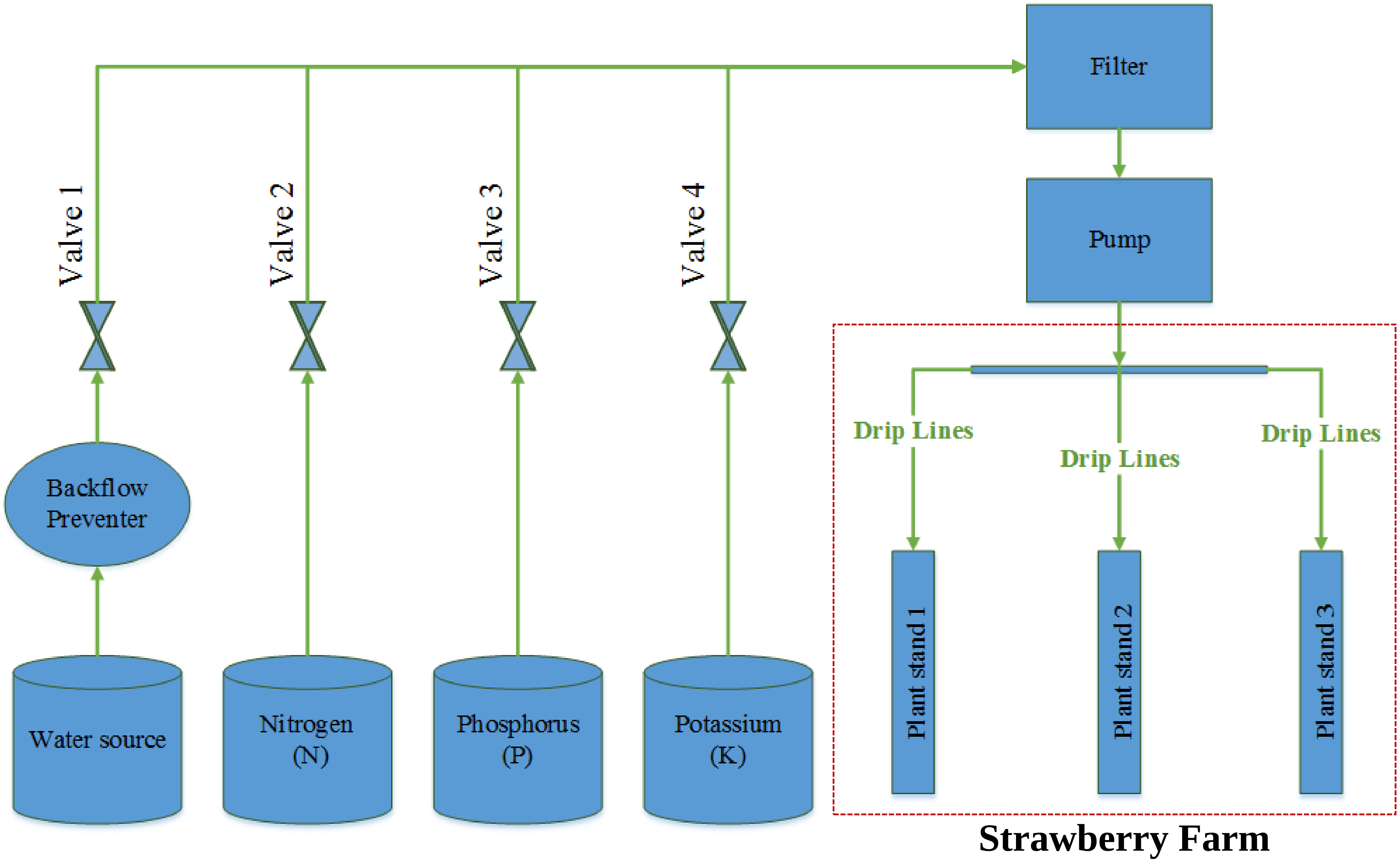
Leaf’s size calculation

-N [5]

-P [5]

-K [5]

Drip Irrigation System



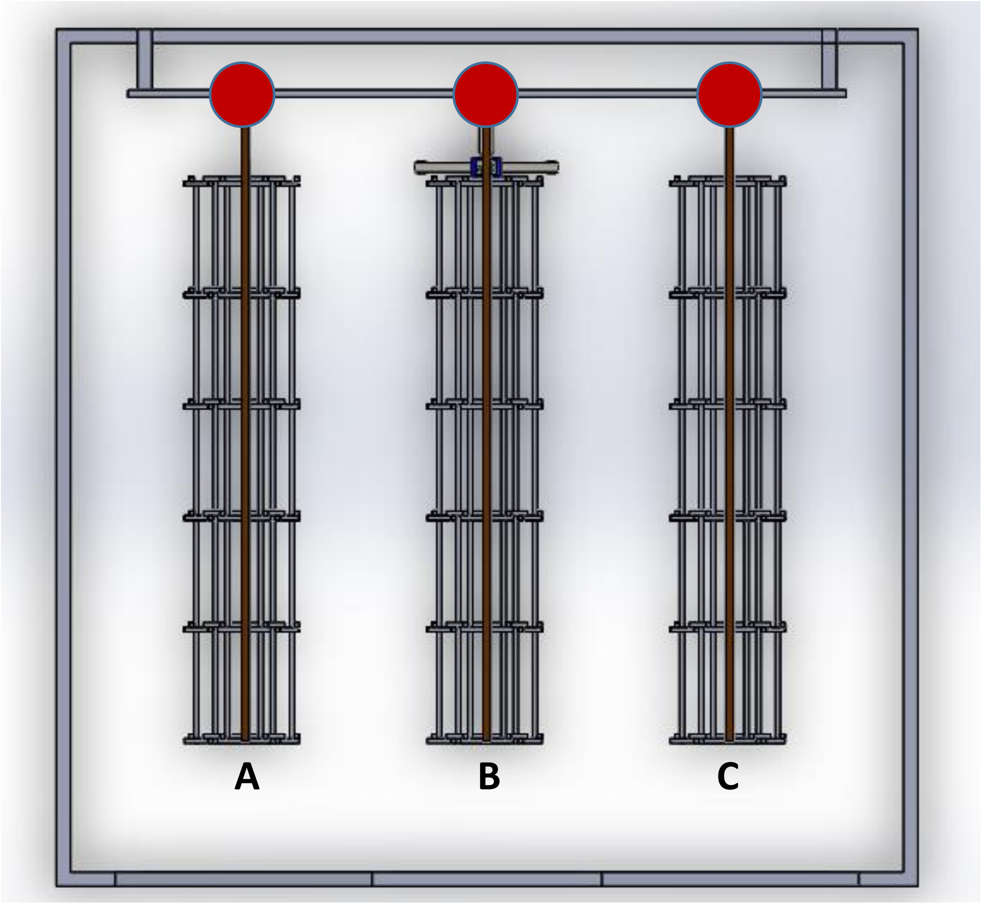
Small-scale Farm Structure and Design

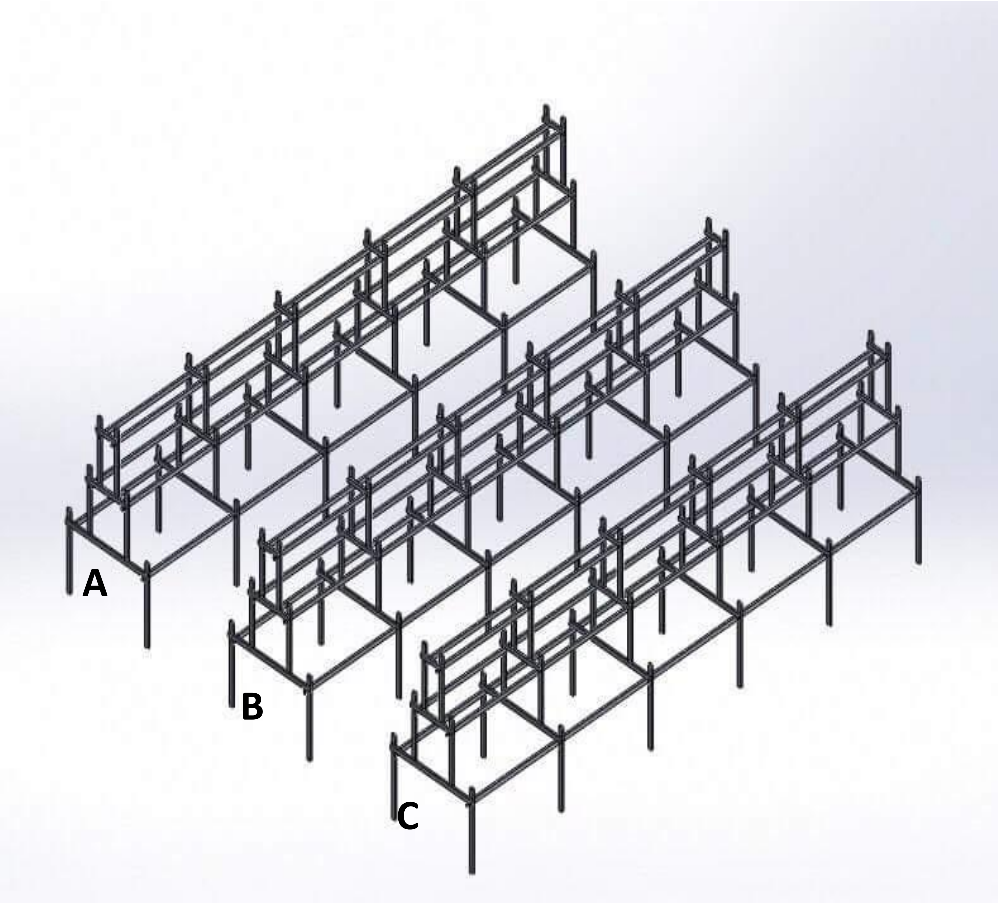
Setting Devices in the Farm

Processes	Requirements
Camera Setting	Logitech C310 Webcam
Drip Kits Setting	DC Pump, Solenoid valve, 24V Power Supply, Offtake with rubber, LDPE pipe, Male Thread Adaptor, 12 Mil Dripline, Flushing Valve
Temperature Sensor Setting	DHT22, Raspberry Pi 3, Real Time Clock

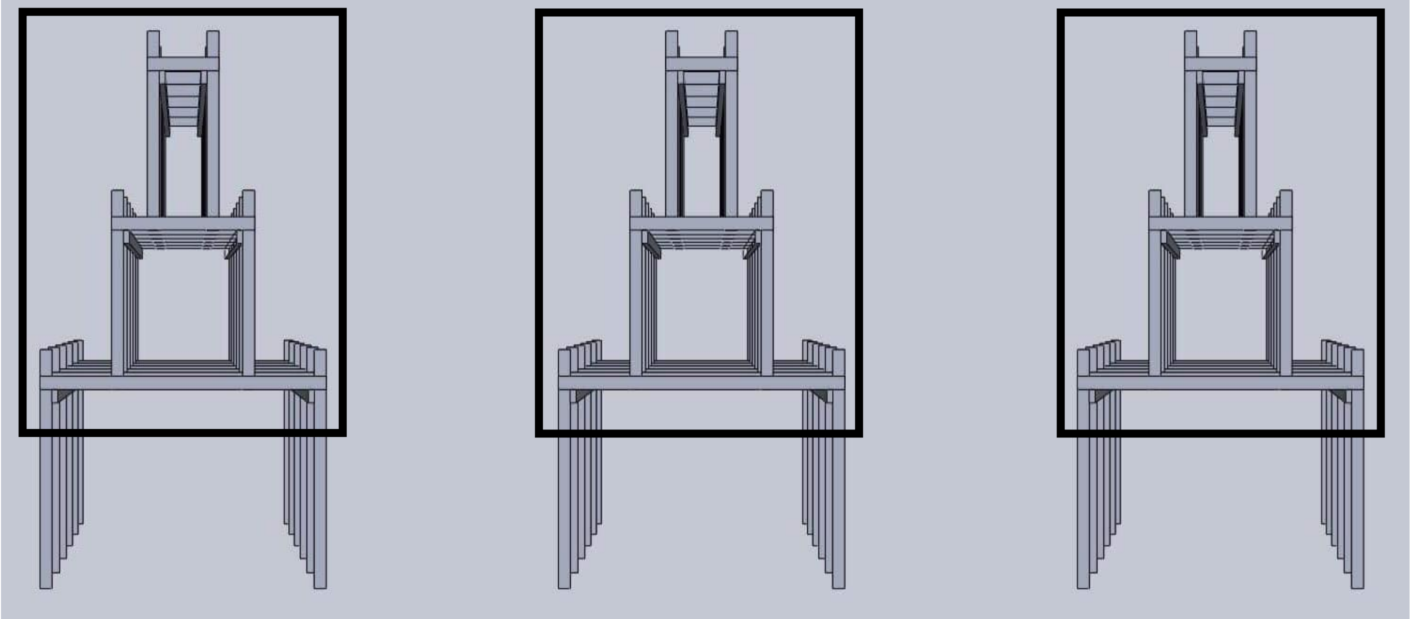
Camera Setting

Camera A Camera B Camera C





Drip Pipes Setting

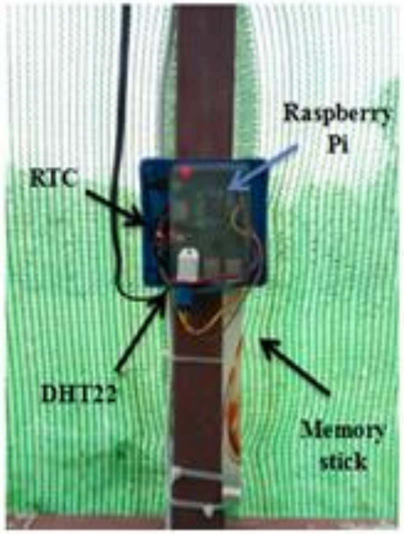


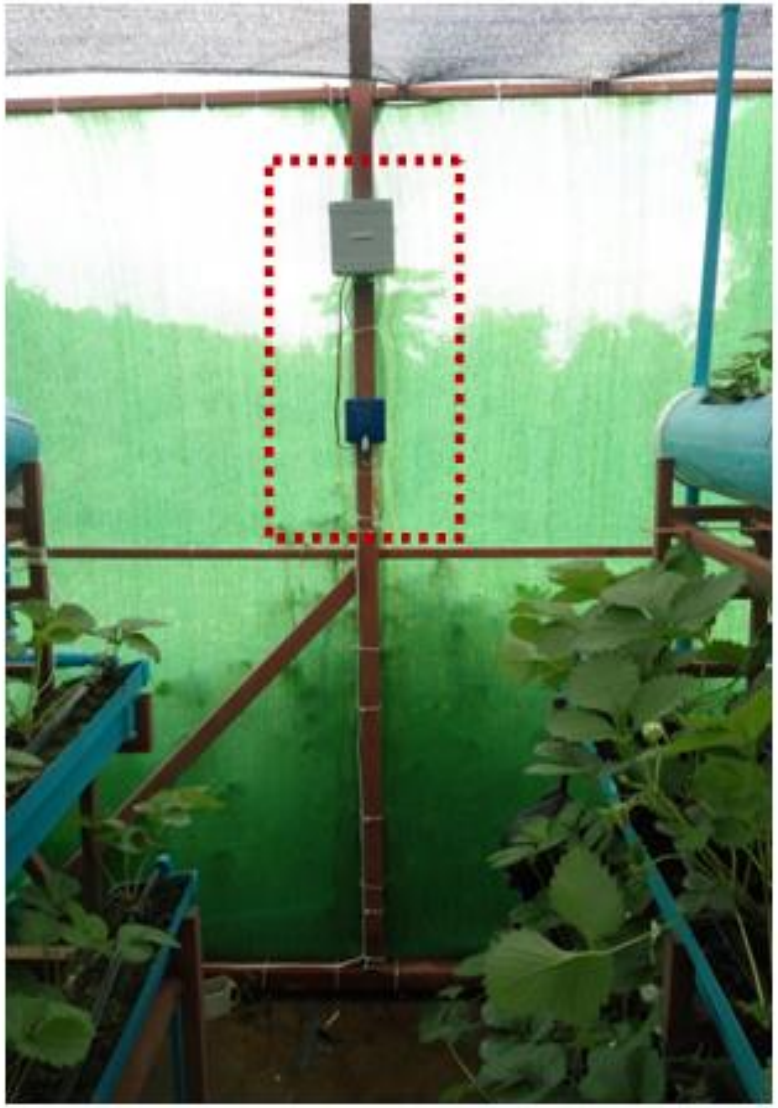
(a) Offtake with rubber, (b) LDPE pipe, (c) Male thread adaptor, (d) 12 mil dripline, (e) Flushing valve

Drip pipes are placed beside the plants around root zones.

Temperature and Humidity Sensor Setting

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND





- The RPi, DHT22 and RTC are installed.
- The temperature and humidity data are logged in the memory stick.

Nutrient Deficiency Symptoms Detection System

Processes	Requirements
Leaf Size Calculation	Some Mathematical Calculations and OpenCV
NPK Detection System	Image Acquisition, Image Pre-processing, Image Segmentation, Feature Extraction, Image Classification, and their appropriate methods
Fertigation (Fertilizer + Drip Irrigation)	Nutrient Solutions for Nitrogen, Phosphorus and Potassium

Image processing techniques

- **Conversion of RGB to GRAY**
- **Morphological Operations**
  - Erosion (erodes away boundries of foreground object)
  - Dilation (adds an extra layer of pixels to foreground)
  - Opening (removing noise)
  - Closing (closing small holes inside foreground object)
- **Otsu's Thresholding**
- **Pyramid Mean Shift Filtering** (to help accuracy of thresholding step)
- **Gaussian Blurring** (to reduce noise and details in image)
- **Canny Edge Detection**
- **Watershed Segmentation**

<u>Python Libraries</u>

- OpenCV
- Matplotlib
- Numpy
- Scipy
- Pandas
- Scikit-learn

How to calculate leaf size?

- Use OpenCV and Python and some mathematical calculation.
- To measure the size of the leaf with OpenCV, a reference object is needed.
- Measures the number of pixels per metric from reference object and then determines the size of leaf.

❖ **Pixels_per_metric = Object_width(measured in pixels) / known_width(measured in metric)**

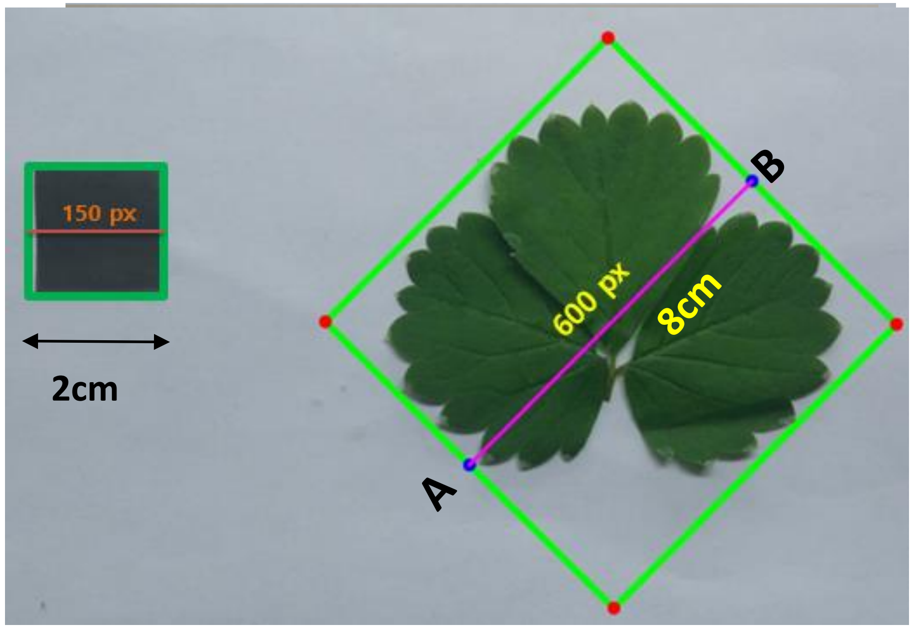
❖ **Dimension of leaf = leaf_width / Pixels_per_metric**

- Known_width = 2cm
- Object_width = 150 pixels

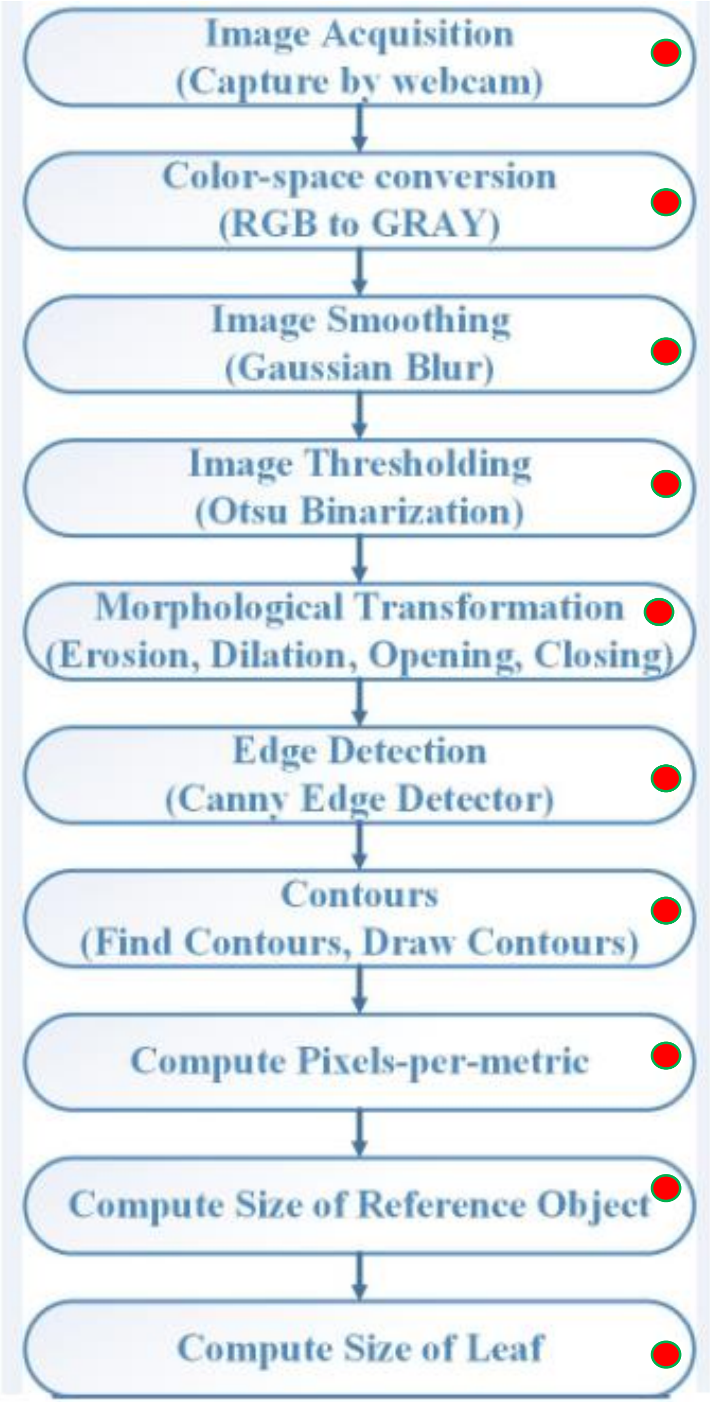
Pixels_per_metric = 150 pixels / 2cm = 75 pixels per cm

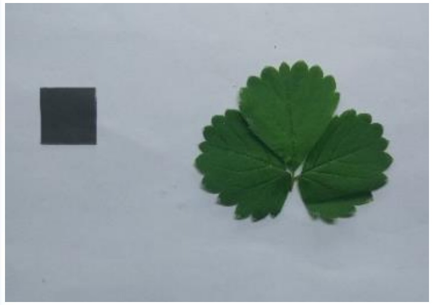
- Leaf_width = 600 pixels

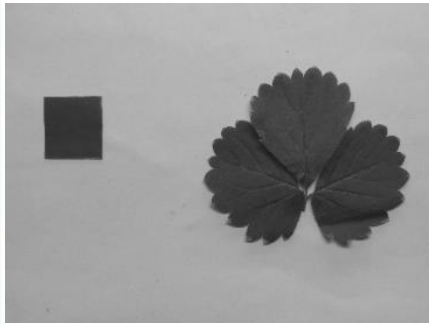
Dimension of leaf = 600 pixels / 75 pixels per cm = 8cm

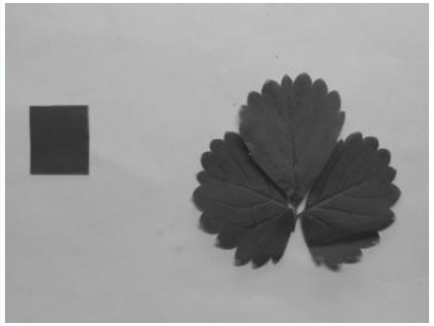


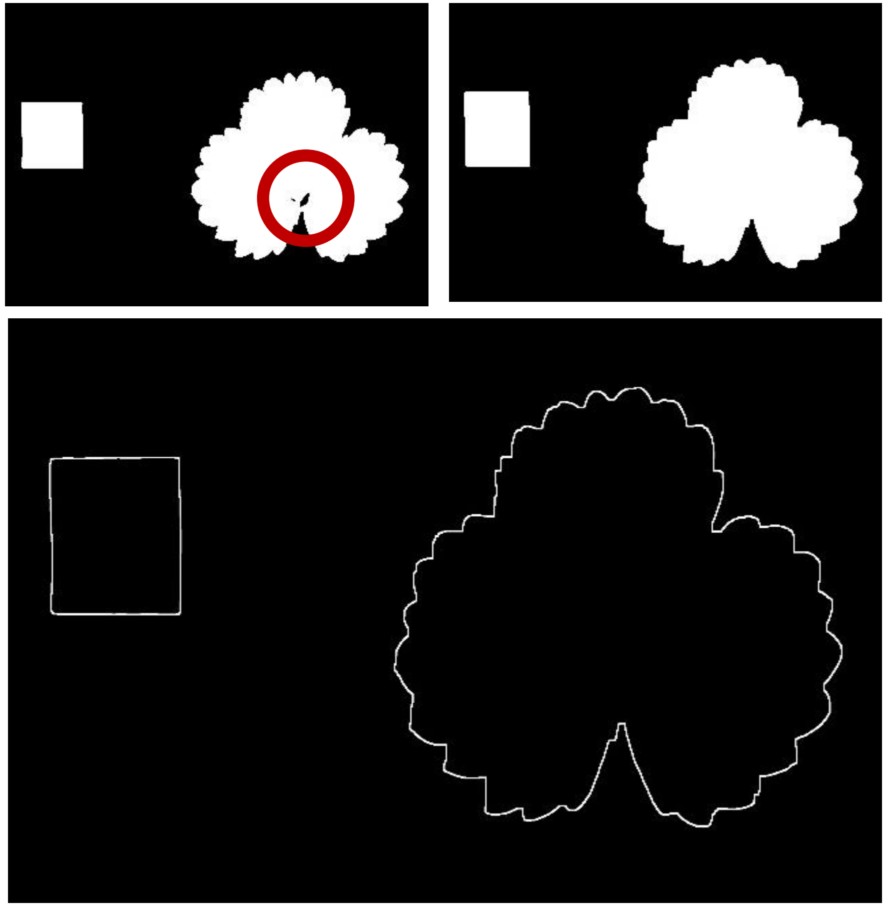
Leaf size calculation by OpenCV

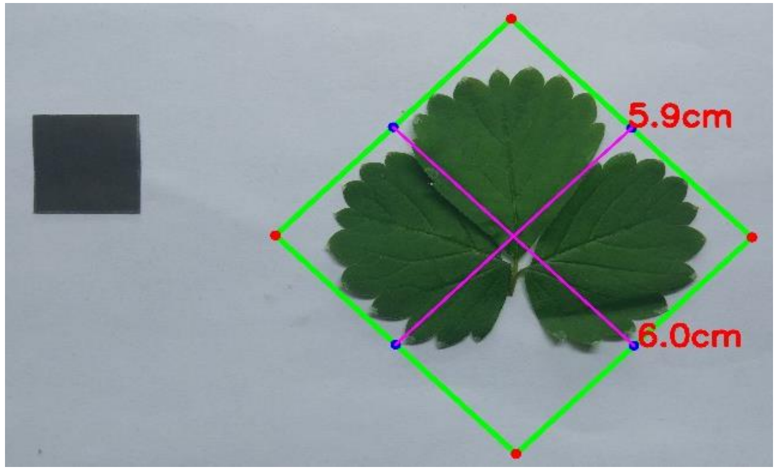




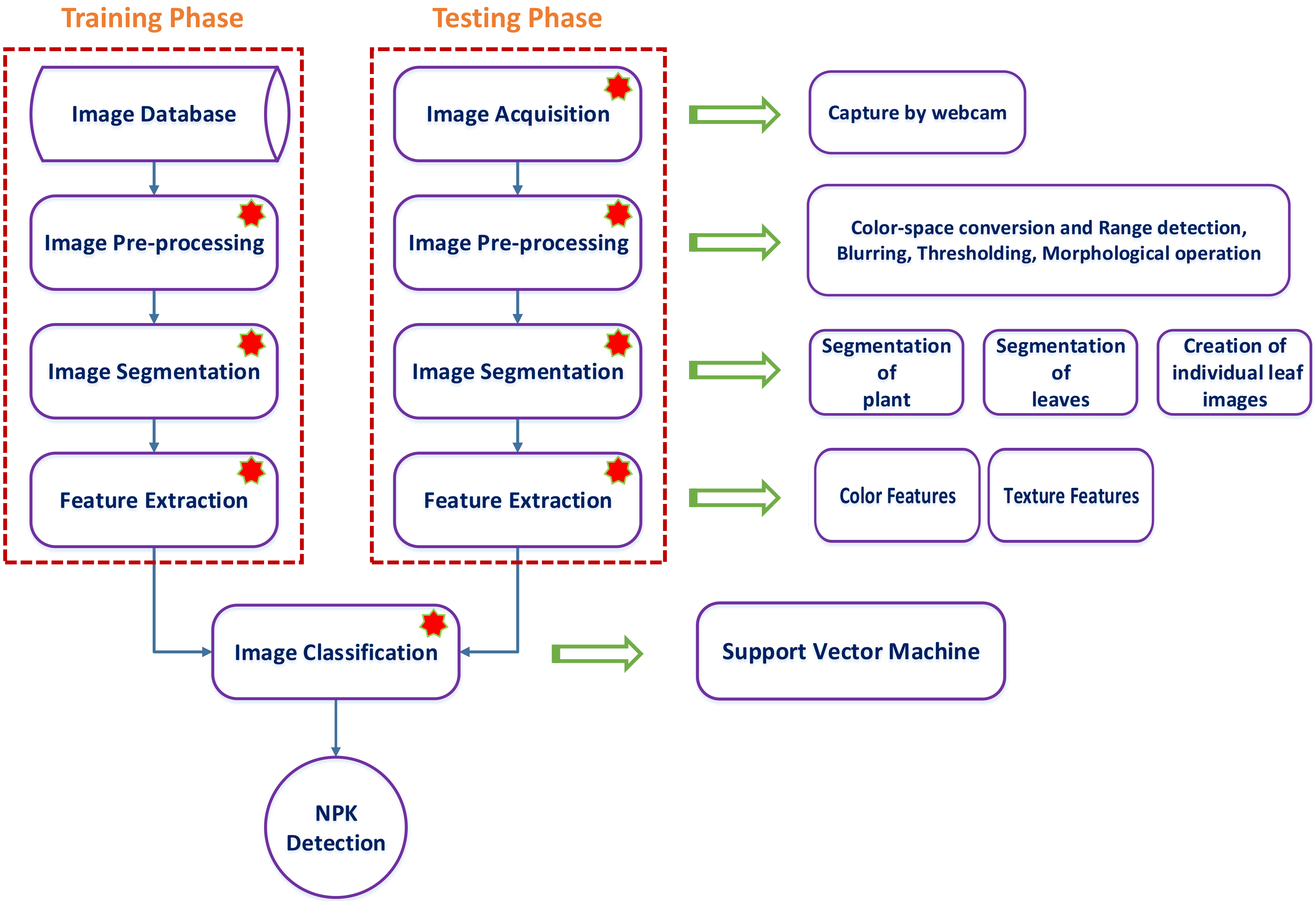


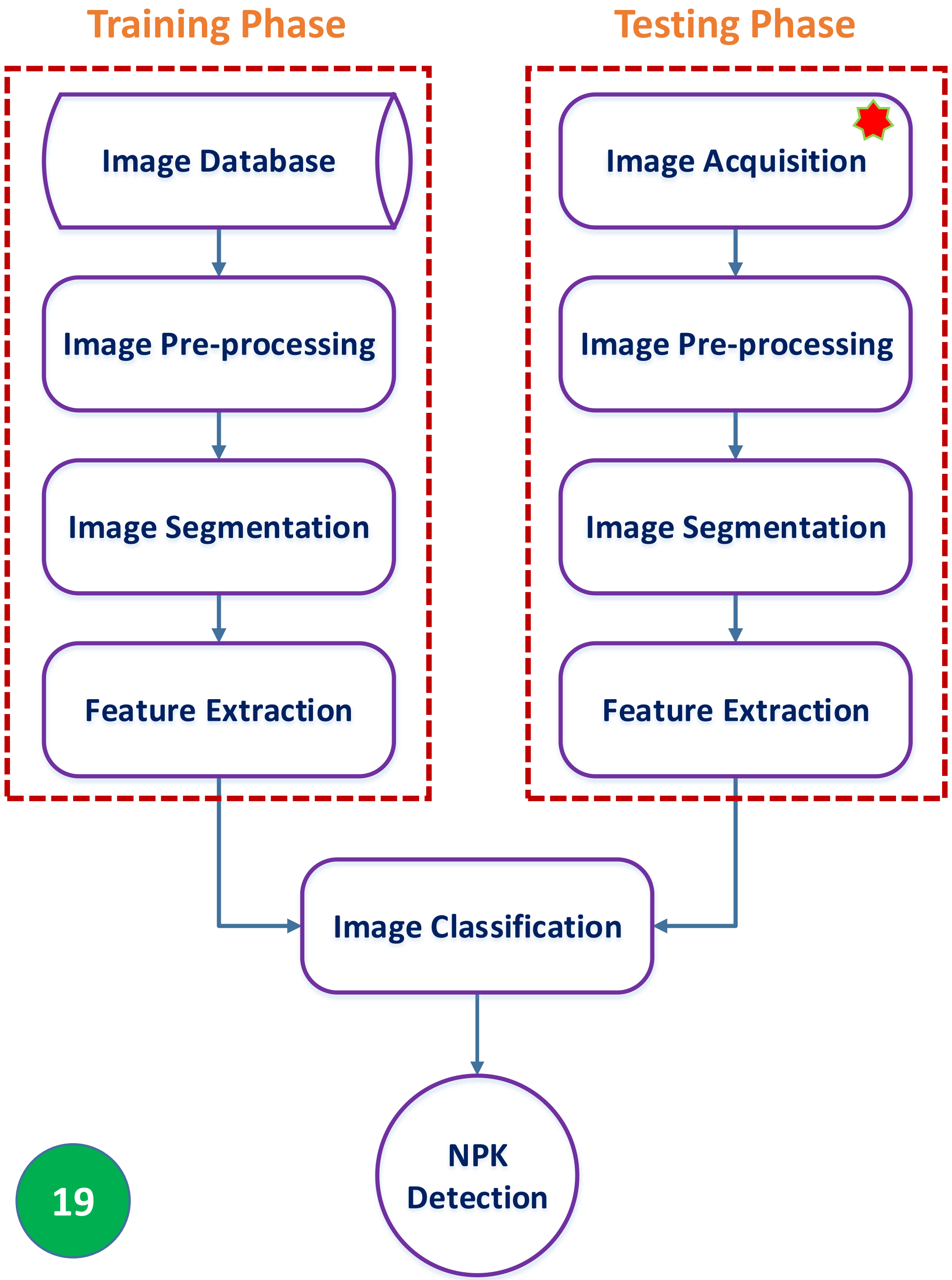






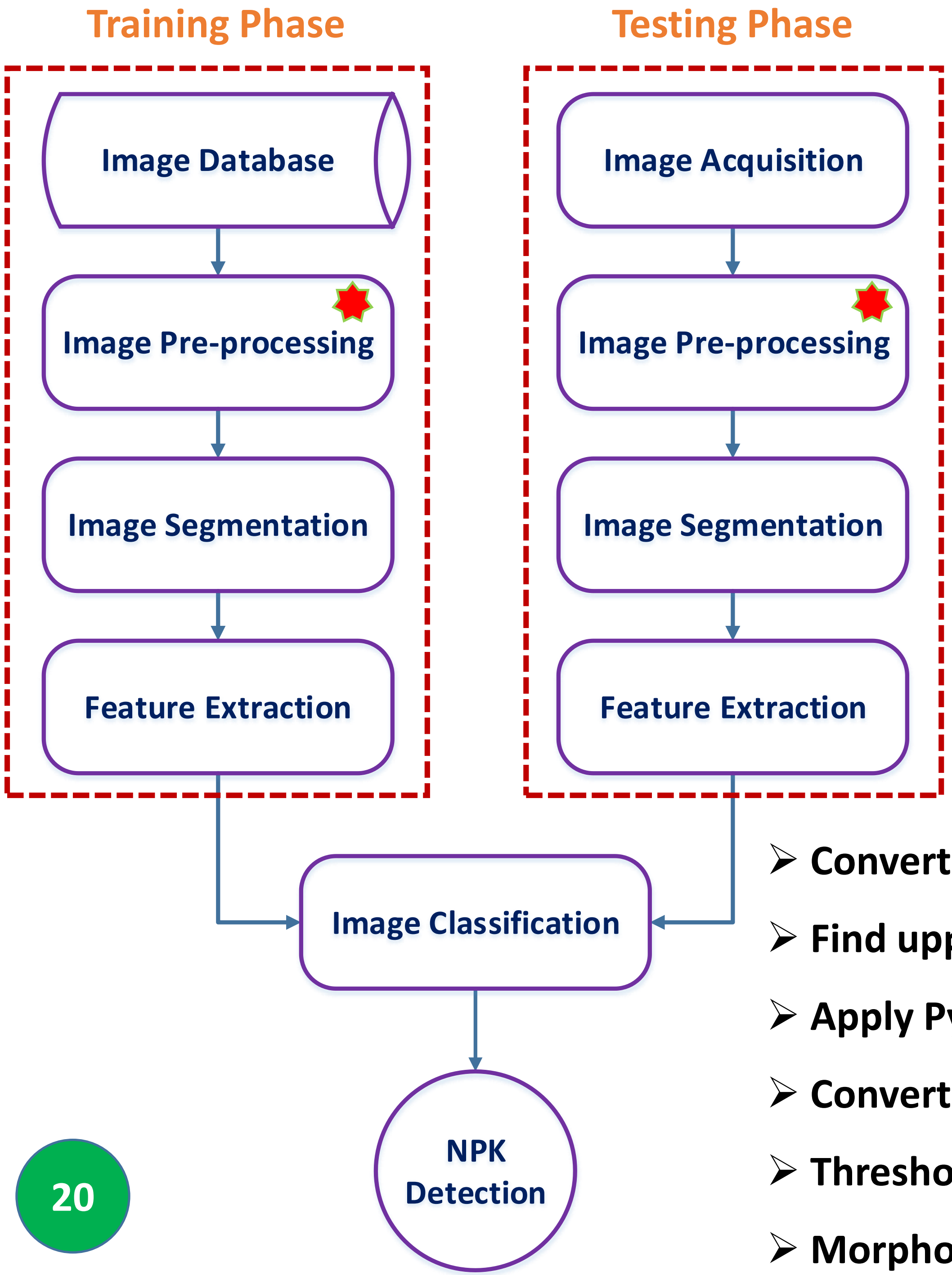
NPK Deficiency detection





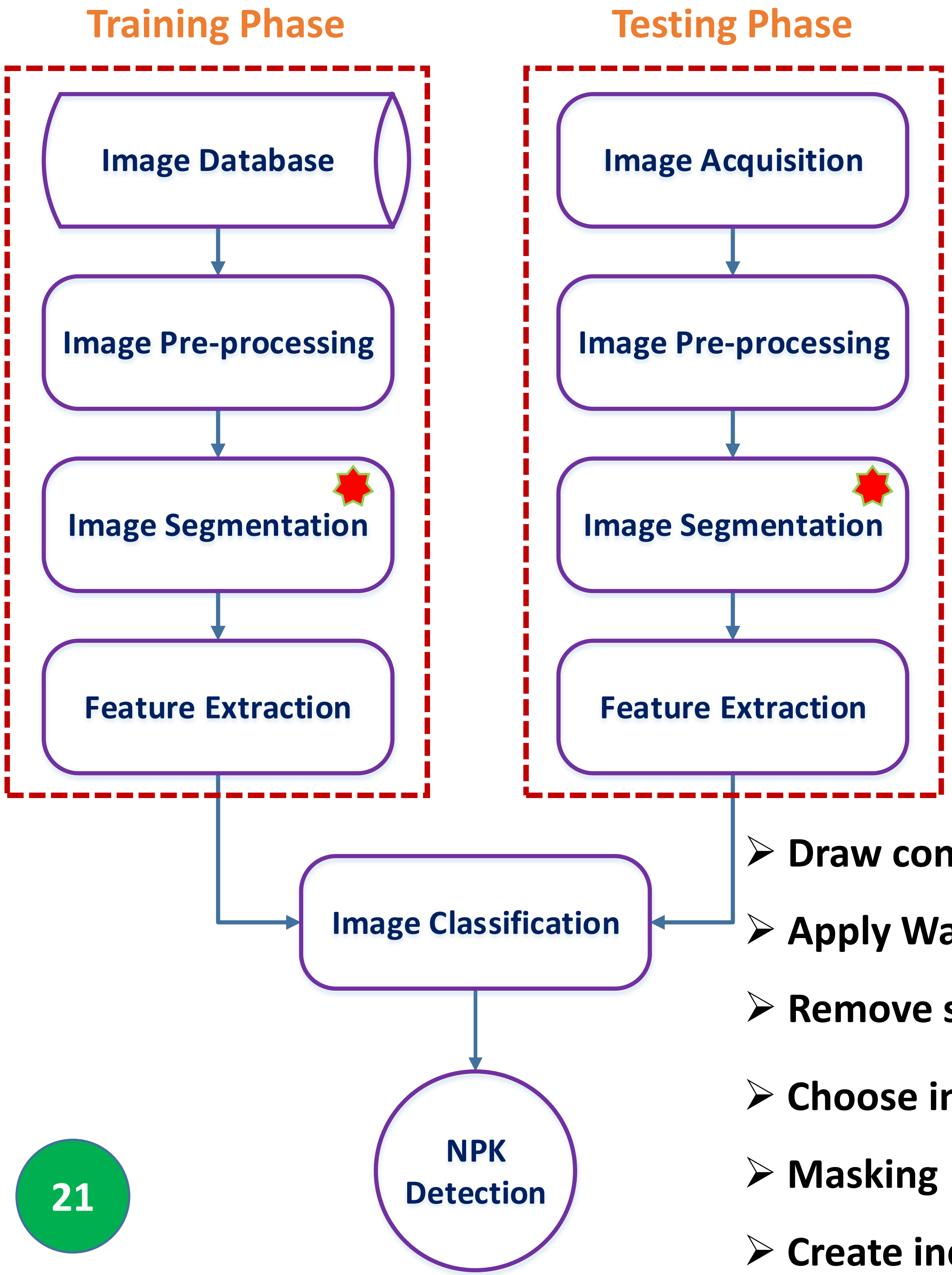
Capture by webcam

- **Load the image**



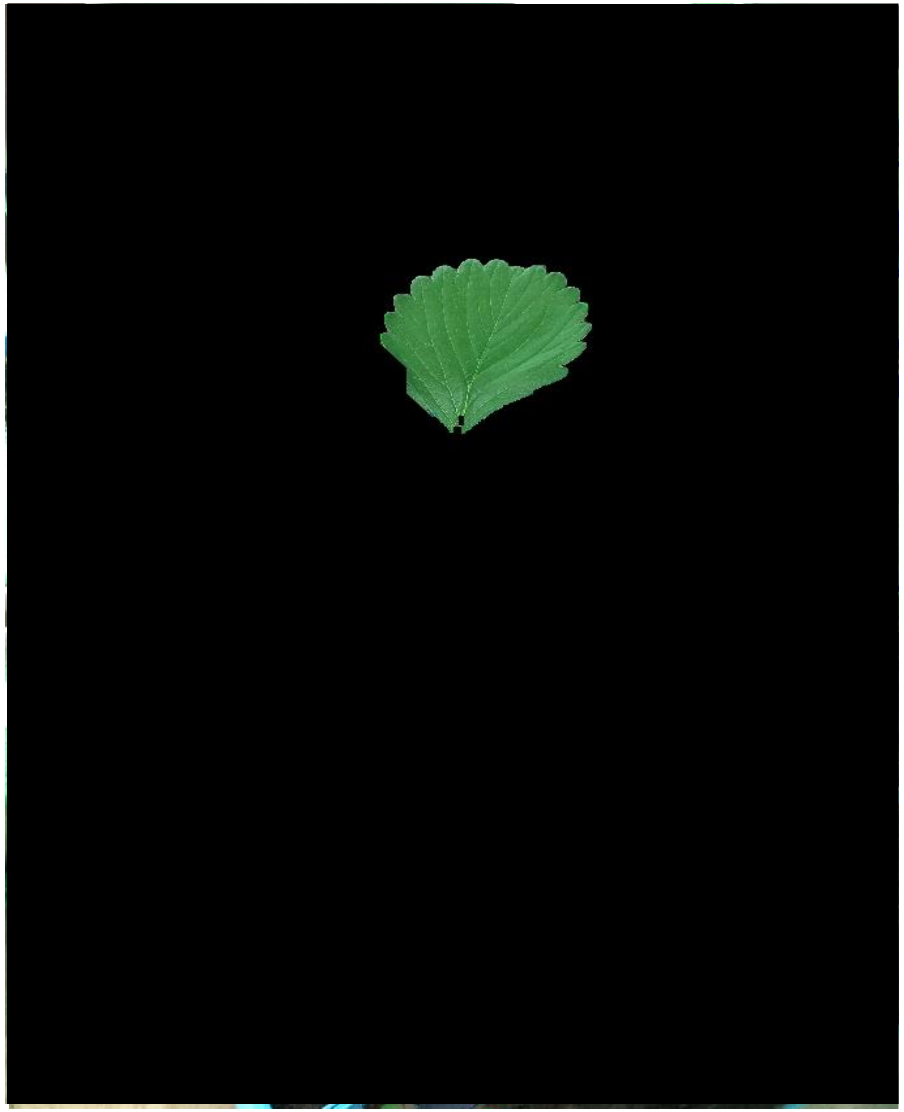
Color-space conversion and Range detection, Blurring, Thresholding, Morphological operation

- Convert RGB image to HSV
- Find upper and lower boundaries
- Apply PyramidMeanShift filtering
- Convert to grayscale
- Thresholding (Binary + Otsu)
- Morphology (Opening + Closing)



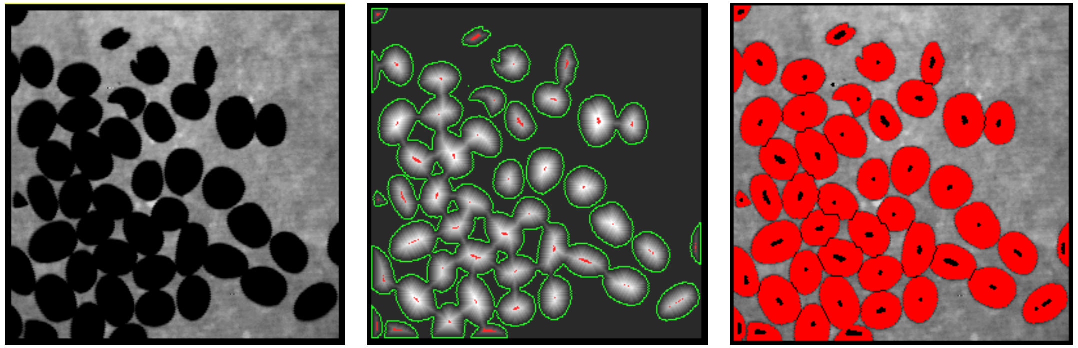
Segmentation of plant | **Segmentation of leaves** | **Creation of individual leaf images**

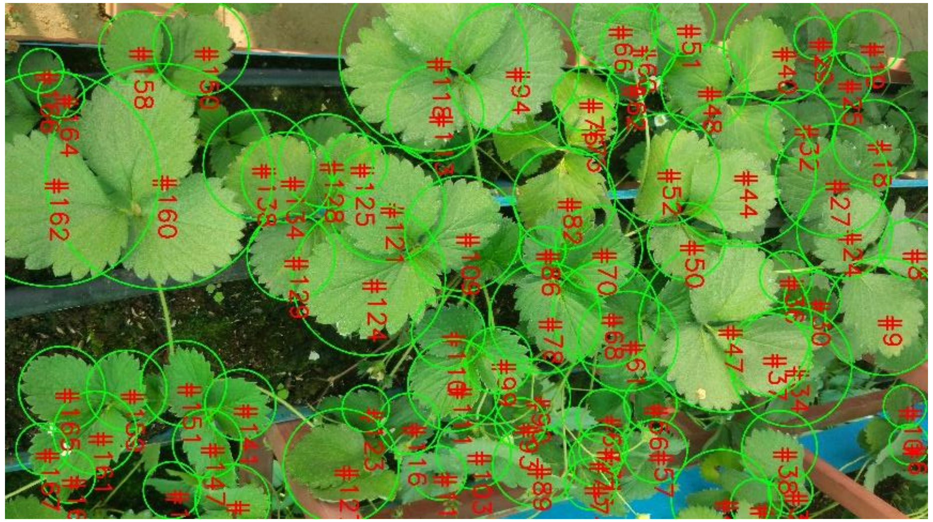
- Draw contours
- Apply Watershed Algorithm
- Remove some small contours area
- Choose individual label
- Masking
- Create individual leaf image



Watershed Segmentation

- A classic algorithm used for segmentation.
- Useful when extracting *touching* or *overlapping* objects.
- Able to detect and extract each leaf.



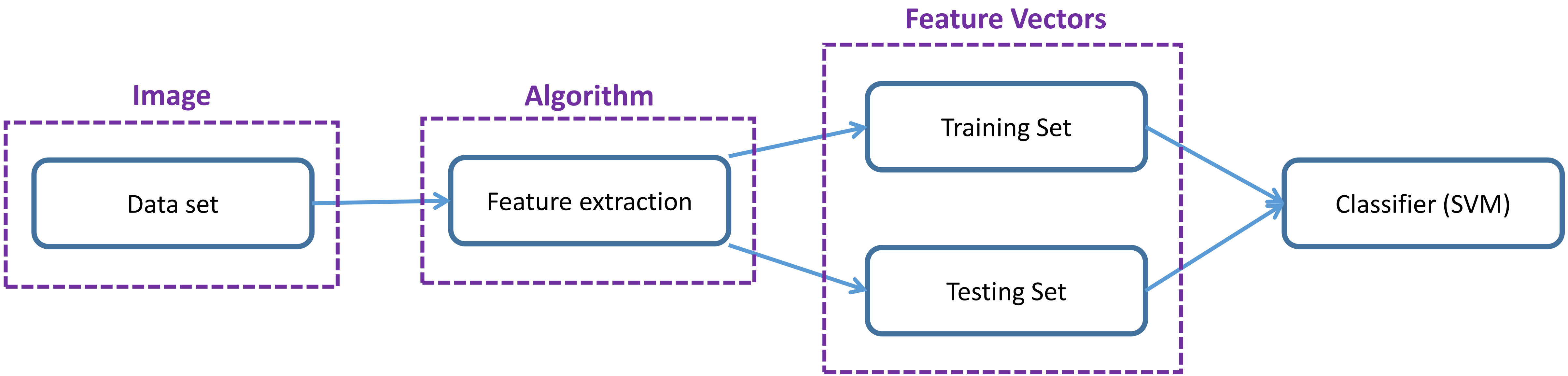


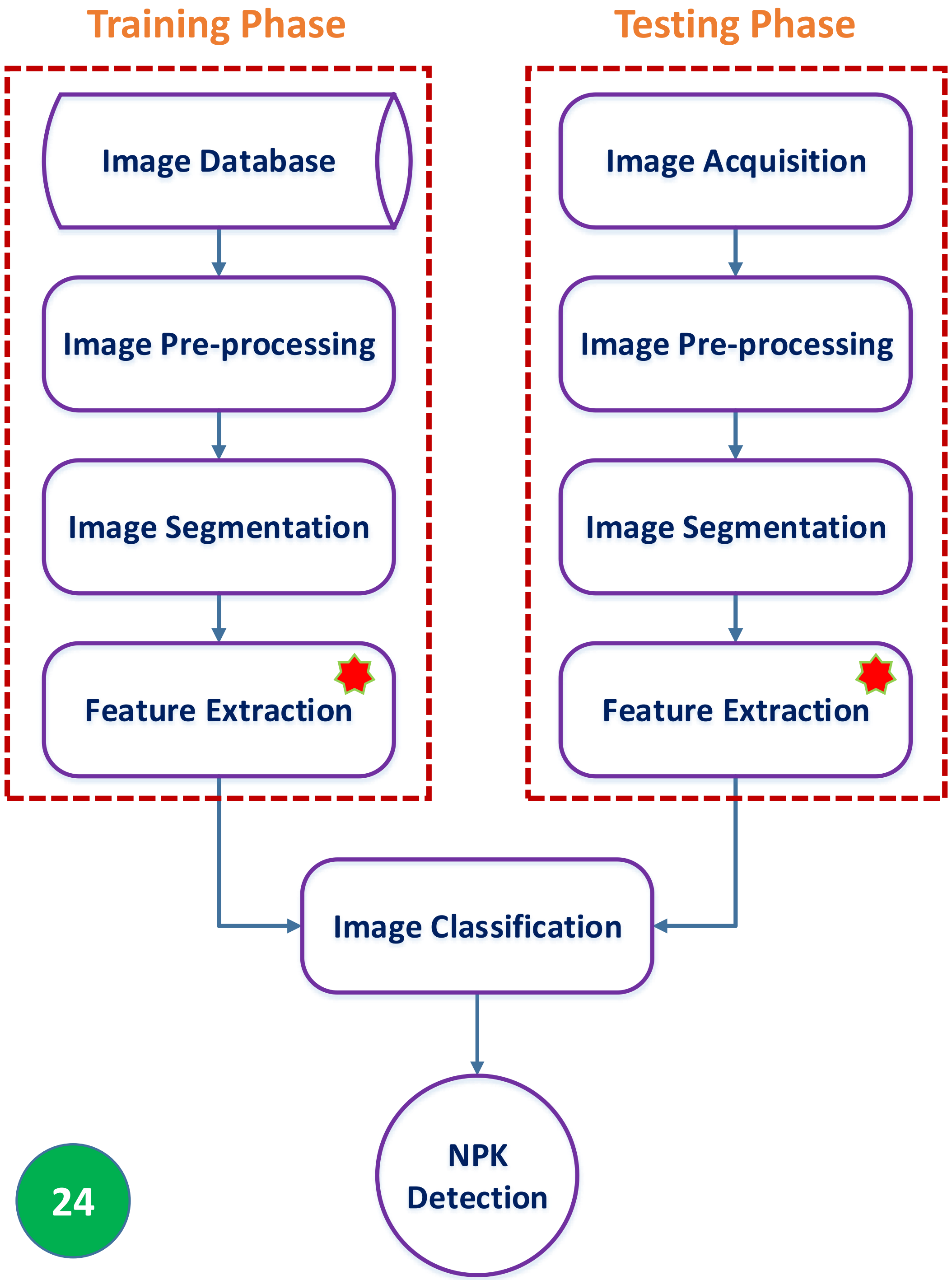
Feature extraction

The images can be described as -

- Spatial (locations, spatial information, etc..)
- Color (RGB, HSV, YCrCb, etc..)
- Texture (rough or smooth, vertical or horizontal, etc..)

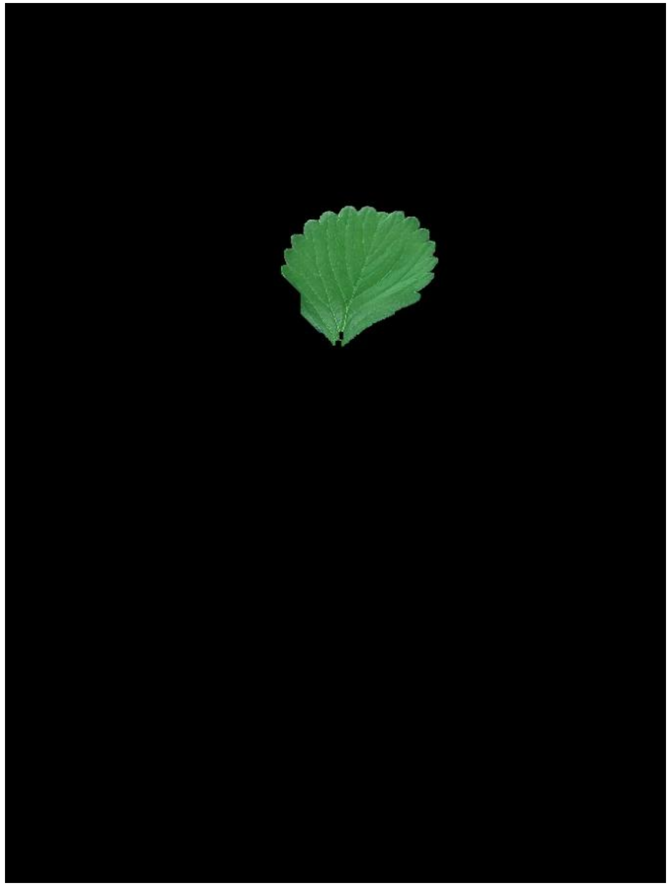
- To encode all this information is to apply feature extraction to quantify the contents of an image.
  - ✓ Taking an input image
  - ✓ Applying an algorithm
  - ✓ Obtaining a feature vector (i.e: a list of numbers) that quantifies images.





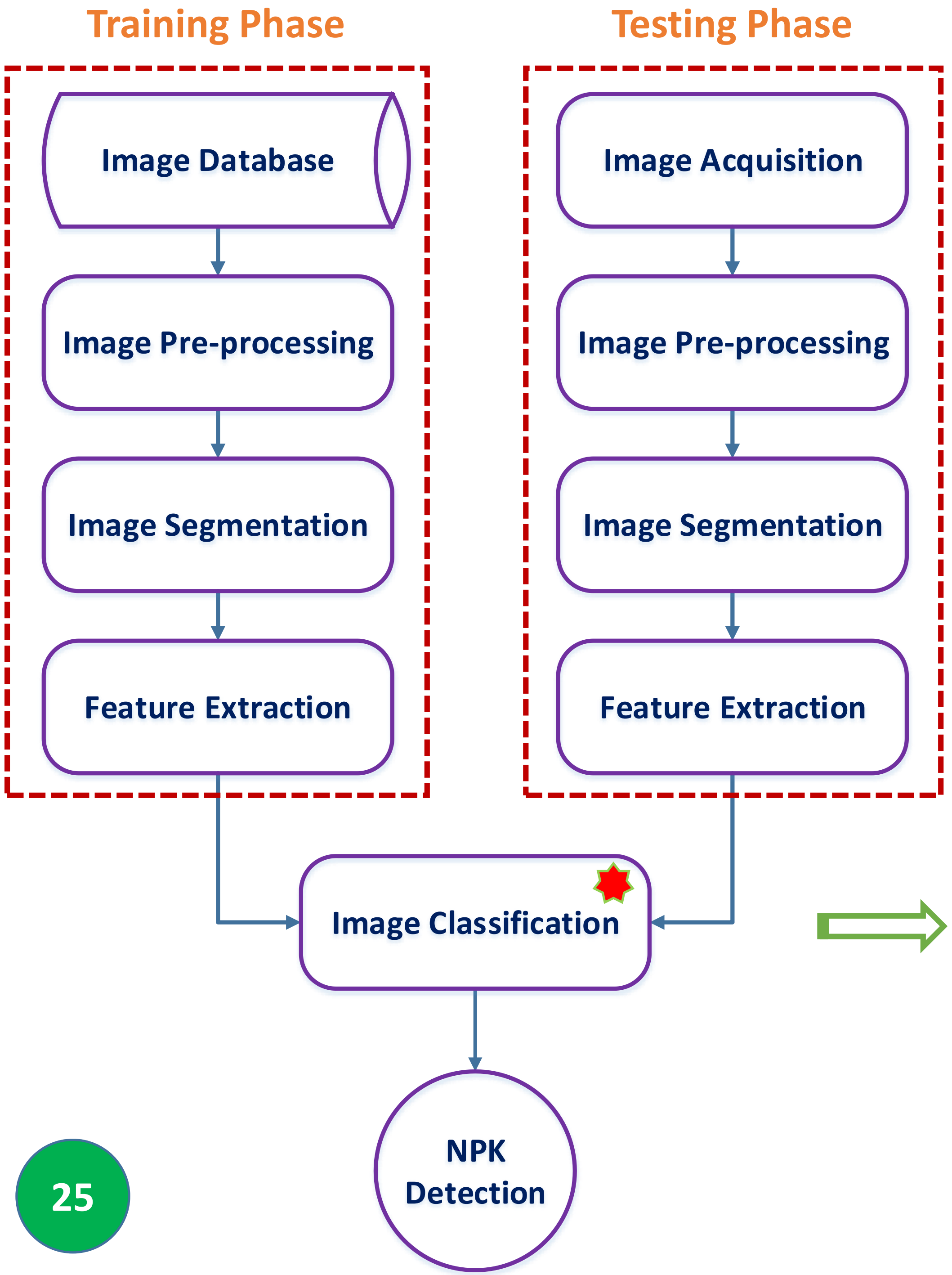
Color Features

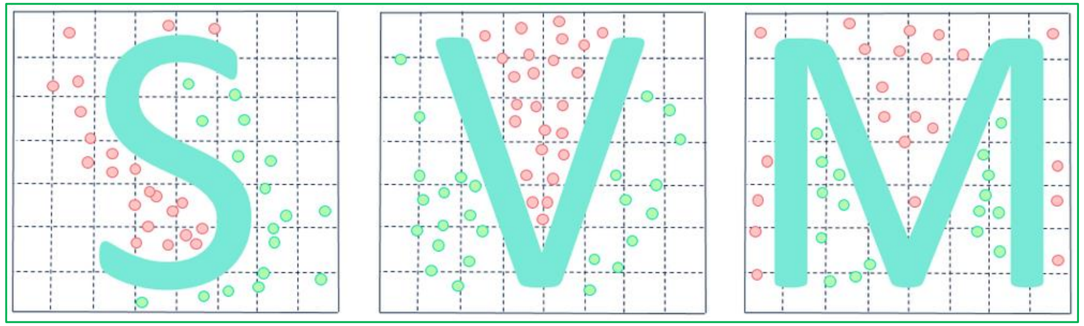
Texture Features



```
----------Color Based Features----------
 Red_mean: 1.7968579727564102
 Green_mean: 3.6438014669625245
 Blue_mean: 1.983820959689349
 Red_std: 11.450839116608064
 Green_std: 22.714943883790514
 Blue_std: 12.639221208153698


----------Texture Based Features----------
 Contrast: 188.19458936630895
 Correlation: 0.9710364964904895
 Inverse_diff_moments: 0.9405118163736426
 Entropy: 4.037106722797929
```





scikit learn

Machine Learning with Scikit-Learn

Support Vector Machine

scikit-learn

Python Machine Learning Algorithm

Image Classification

- The task of assigning a label to an image from a predefined set of categories.
- To analyze an input image and return a label that categorizes the image.
- The labels are always from a predefines set of possible categories.

Healthy N-deficiency P-deficiency K-deficiency

Supervised or Unsupervised?

Healthy

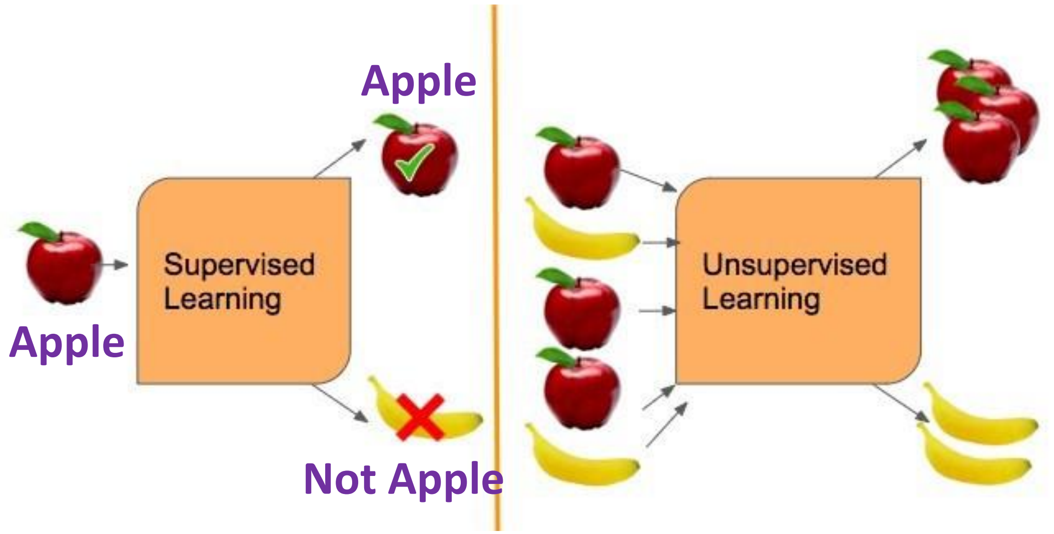
N-deficiency

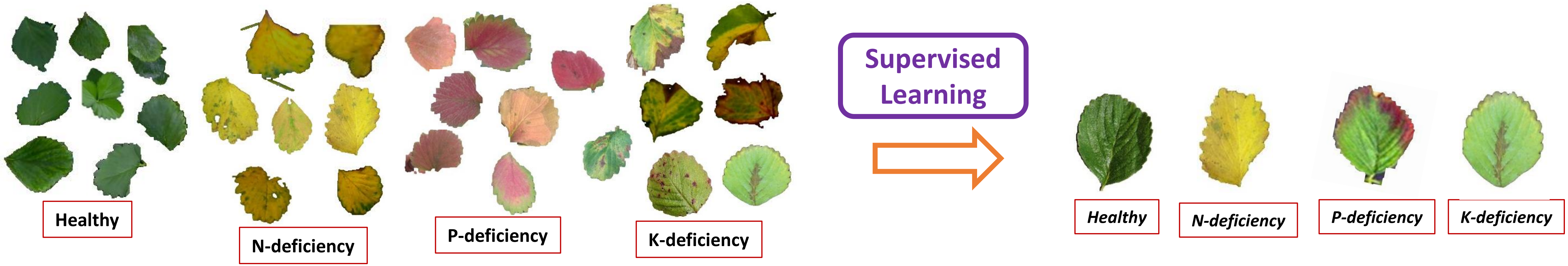
P-deficiency

K-deficiency

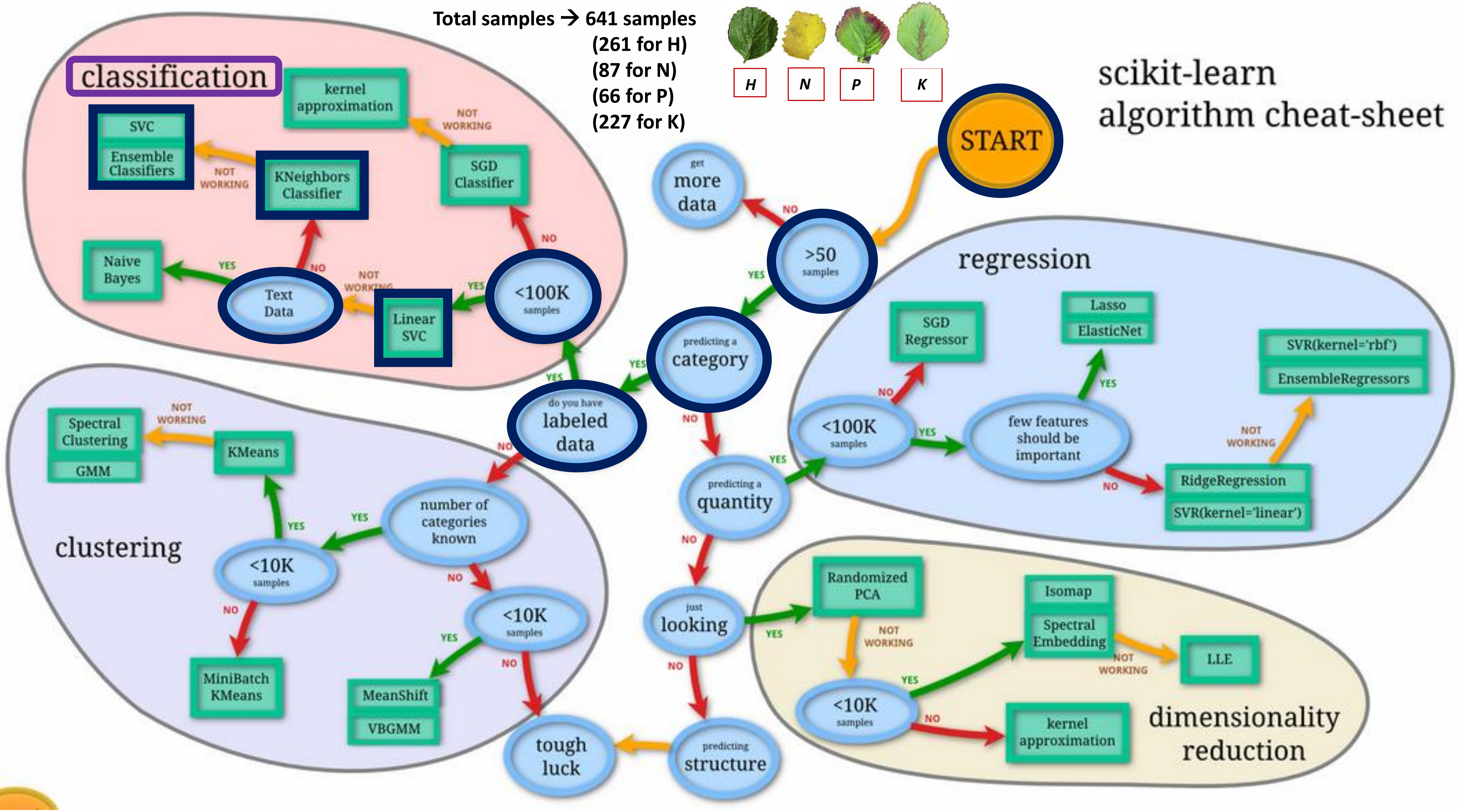
Types of Learning

- **Supervised learning** - the dataset is the collection of **labeled examples** and **has the knowledge of output**.
- **Unsupervised learning** - the dataset is the collection of **unlabeled examples** and **has no knowledge of output.**



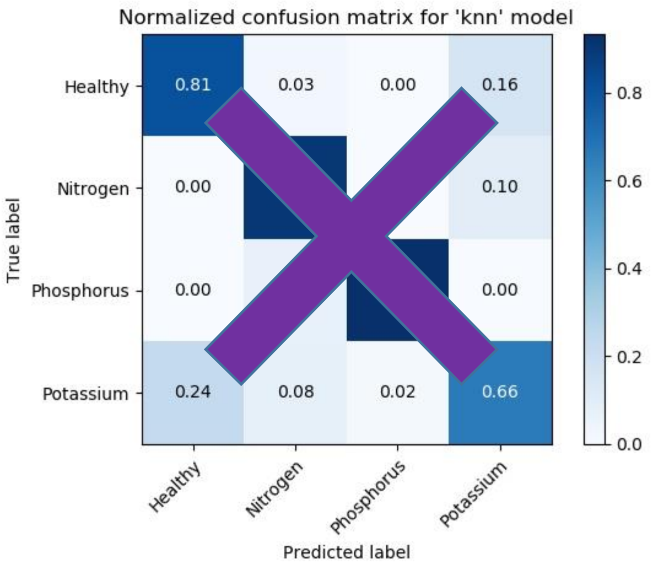


Total samples → 641 samples
(261 for H)
(87 for N)
(66 for P)
(227 for K)

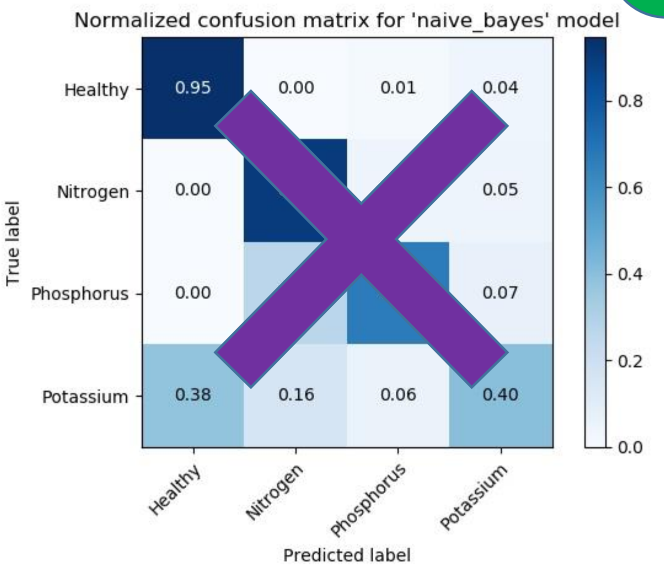


Spot-Check

1. K-Nearest Neighbours (kNN)
2. Naïve_Bayes
3. Support Vector Machine (SVM)
4. Decision Tree
5. Logistic Regression

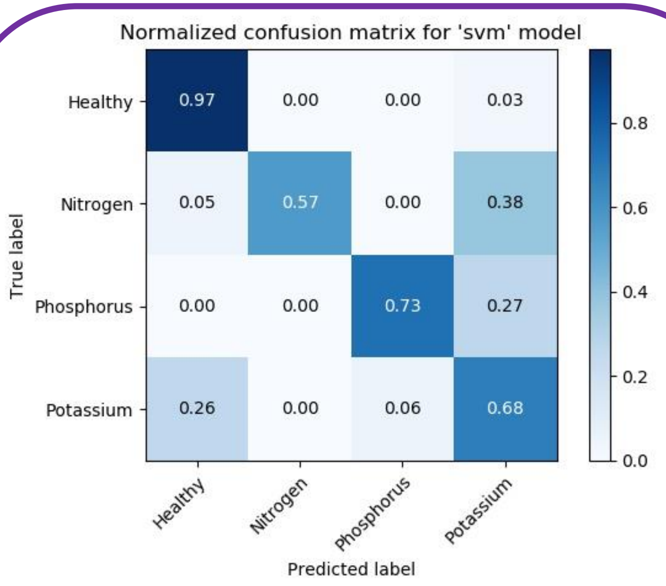


kNN (78.88%)

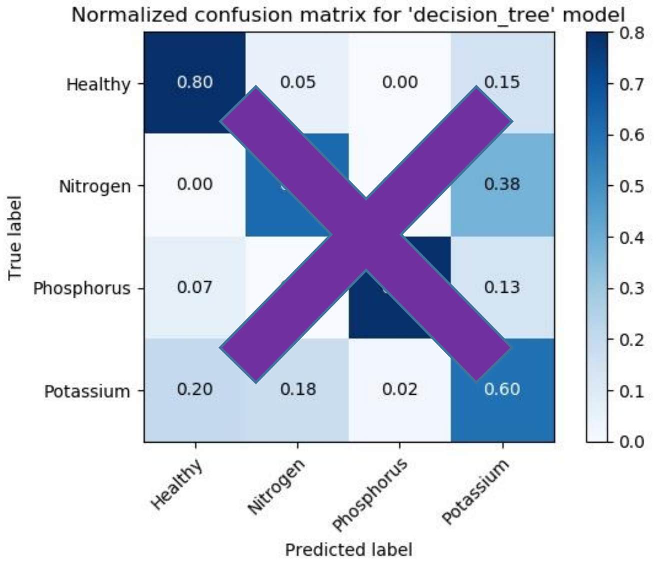


Naïve_Bayes (74.53%)

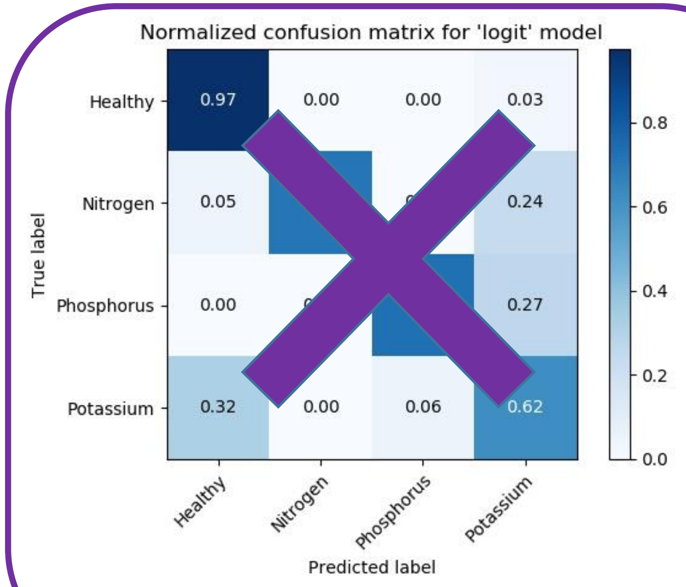
Kernel, C, Gamma



SVM (80.75%)



Decision_Tree (73.29%)



Logistic Regression (80.75%)

Multiclass SVM

- **Multiclass classification** means a classification task with more than two classes.
- Each sample is assigned to one and only one label.
- SVM is able to classify only binary data, need to convert the multi-dimensional dataset into binary form using:

  a. ***One vs Rest***

  b. ***One vs One***

One vs Rest

- Constructs one classifier per class.
- For each classifier, the class is fitted against all the other classes.
- Requires to fit only n_classes classifiers.

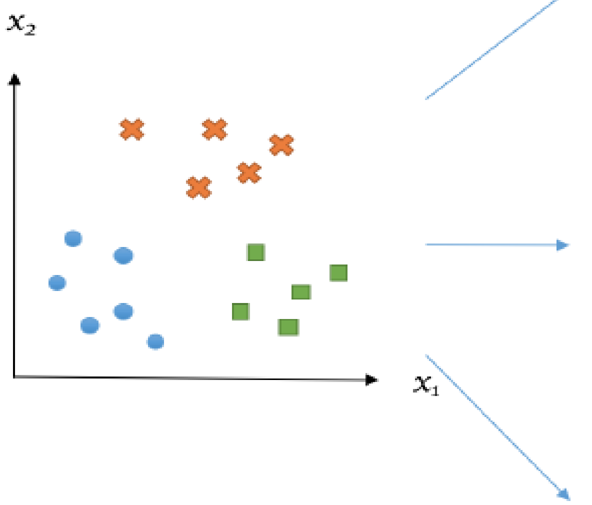
One vs One

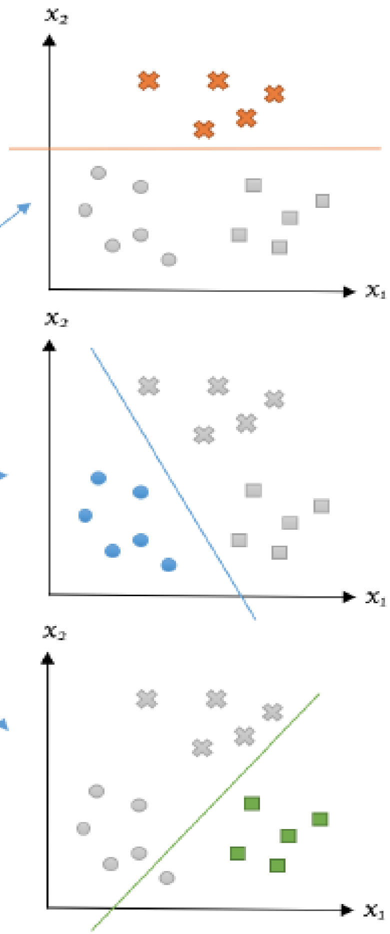
- Constructs one classifier per pair of classes.
- Requires to fit (n_classes * (n_classes - 1) / 2) classifiers.

Multiclass SVM

From the full dataset, construct three binary classifiers, one for each class

OvR : One vs Rest

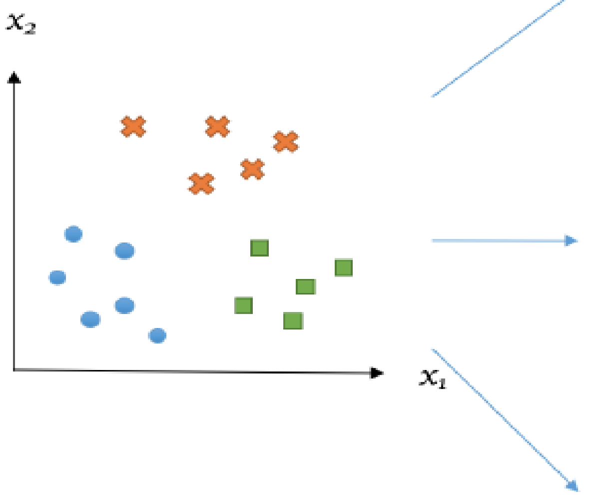


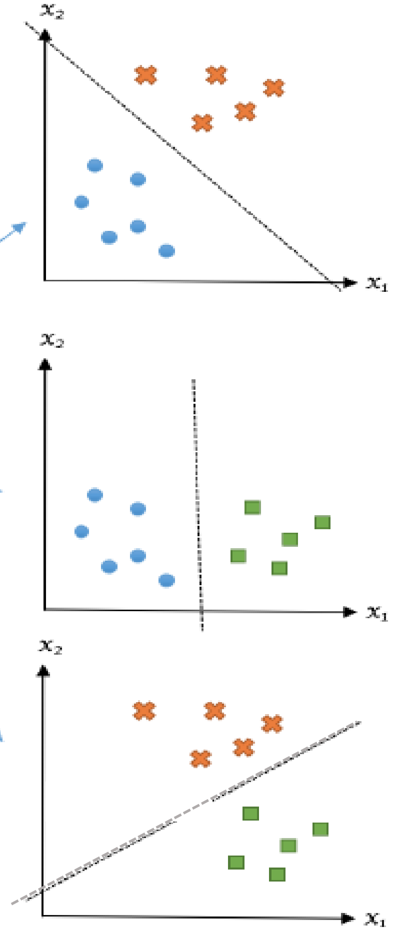


Pros: **Fewer classification**
Cons: **Classes may be imbalanced**

n_classes

OvO : One vs One





Pros: **Less sensitive to imbalance**
Cons: **More classifications**

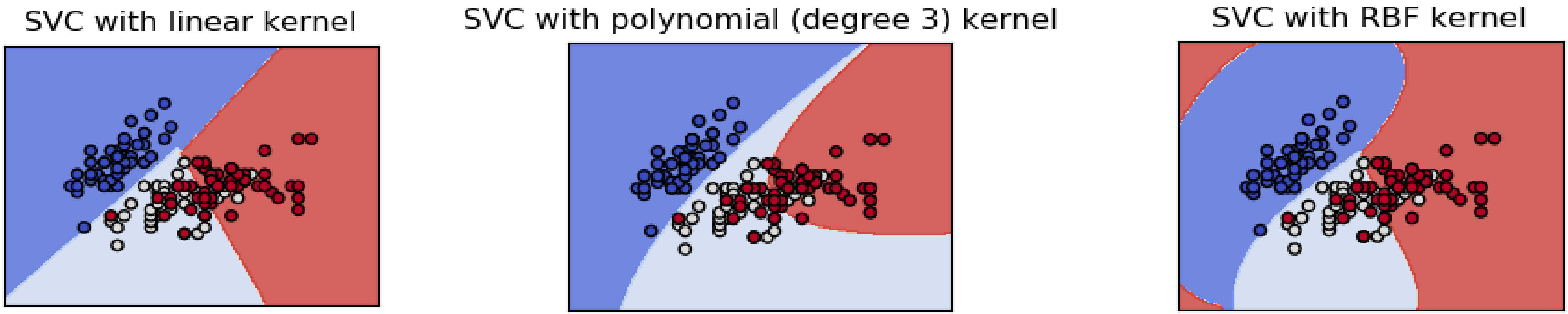
n_classes * (n_classes – 1)/2

Non-linear SVM

Types of kernels [6]

- Linear kernel
- Polynomial kernel
- Radial basis function kernel (RBF)/ Gaussian Kernel

scikit-learn documentation

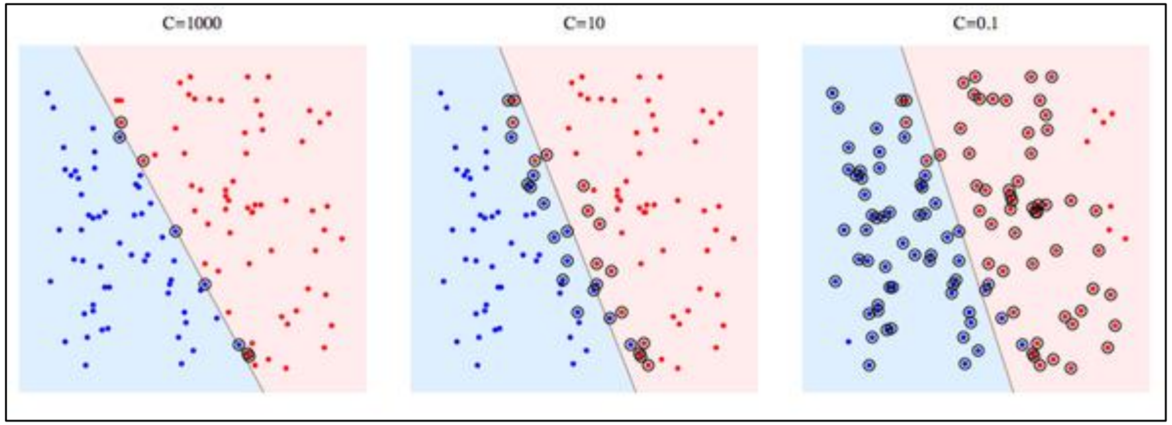


SVM PARAMETERS

- Parameters are arguments that are passed when the classifier is created.

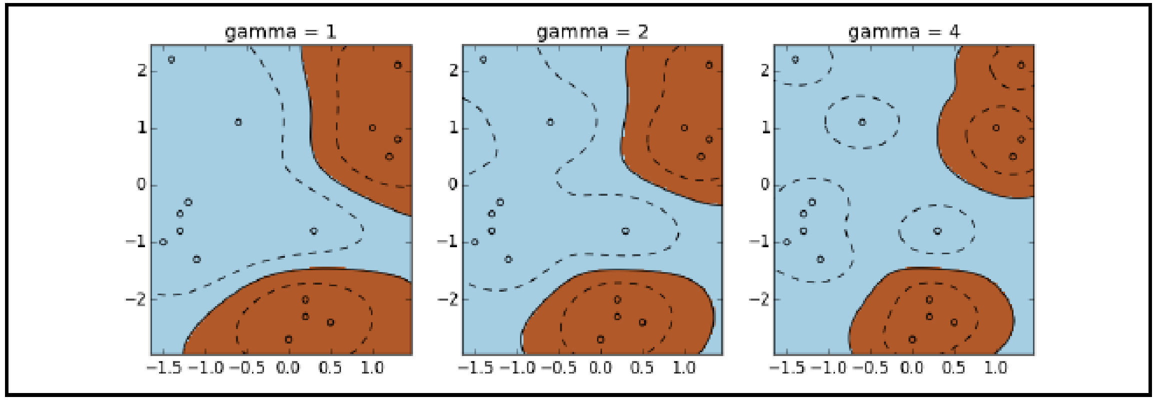
C:

- Controls the trade off between ***smooth decision boundary*** and ***classifying training points correctly***.
- Small C → misclassification low ("soft margin").
- Large C → misclassification high ('hard margin").



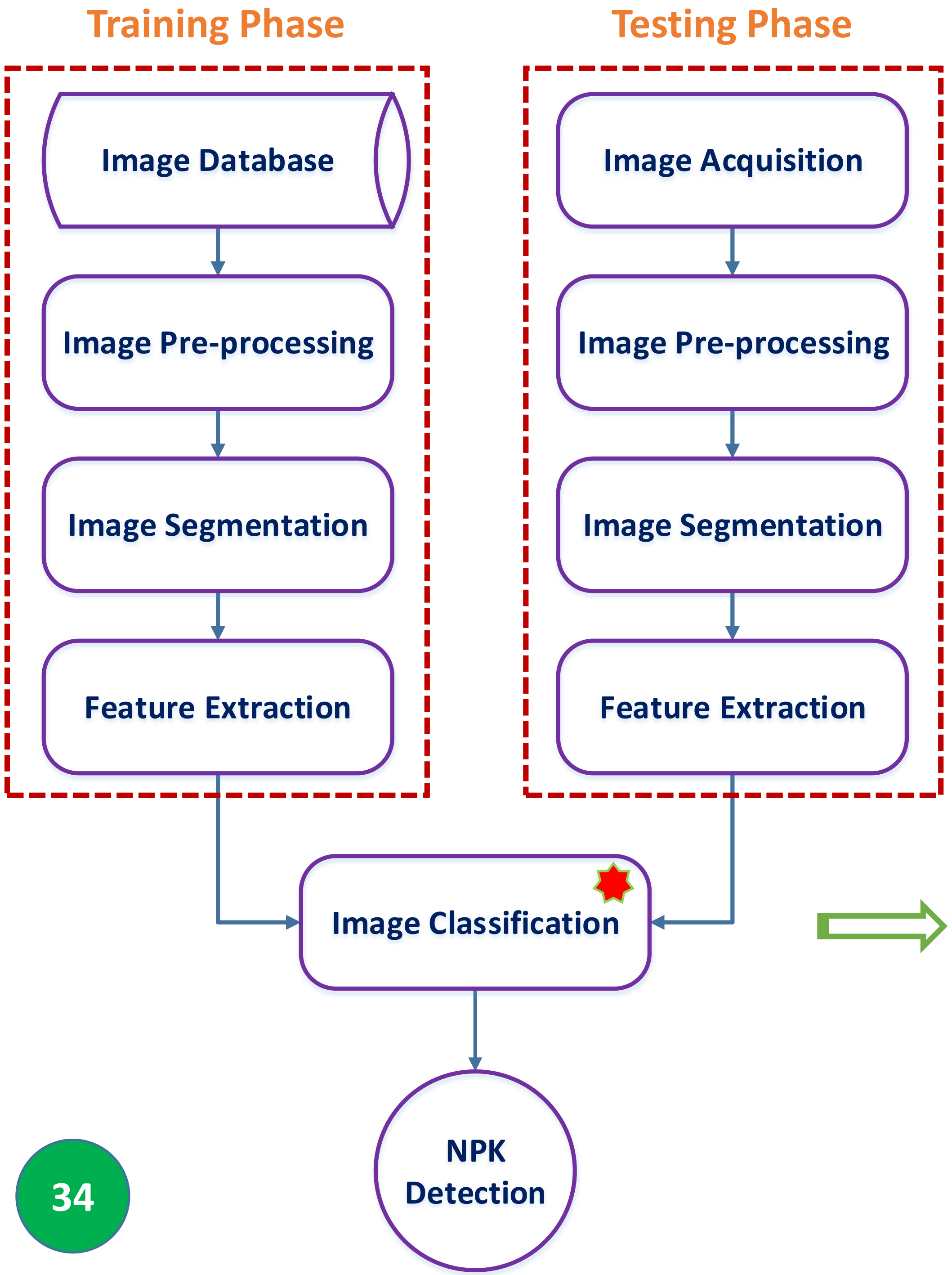
Gamma(used for RBF kernel and SIGMOID kernel)**:**

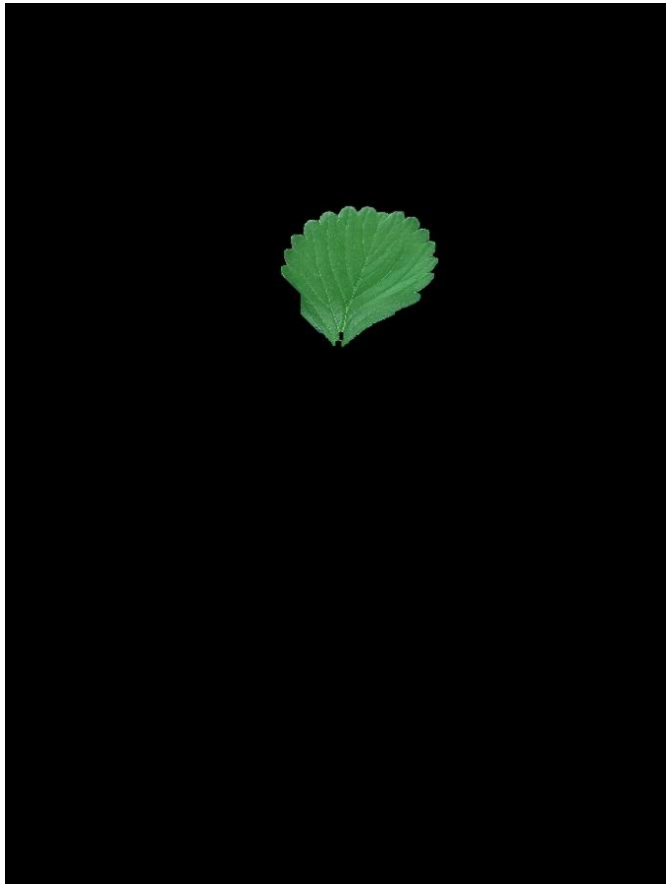
- Defines how far the influence of a single training example reaches.
- Small Gamma → every point has a far reach.
- Large Gamma → every point has close reach.



The goal is to find the balance between "not too strict" and "not too loose".

GridSearchCV method





Classify this leave is a *'Healthy'* leave

Support Vector Machine

scikit-learn

Python Machine Learning Algorithm

Test and Results (Training and Parameter Tuning)

	Total Samples	Training Set (80%)	Test Set (20%)
Healthy	261	209	52
Nitrogen	87	70	17
Phosphorus	66	53	13
Potassium	227	182	45

- **Training set** — a subset to train a model.
- **Test set** — a subset to test the trained model.

Kernel	C	Gamma
RBF	0.001,0.01,0.1,10,25,50,100,1000	1e-2, 1e-3, 1e-4, 1e-5
SIGMOID	0.001,0.01,0.1,10,25,50,100,1000	1e-2, 1e-3, 1e-4, 1e-5
LINEAR	0.001,0.01,0.1,10,25,50,100,1000	-

Decision function shape → 'OvR'
Kernel → 'Linear'
C → 1000

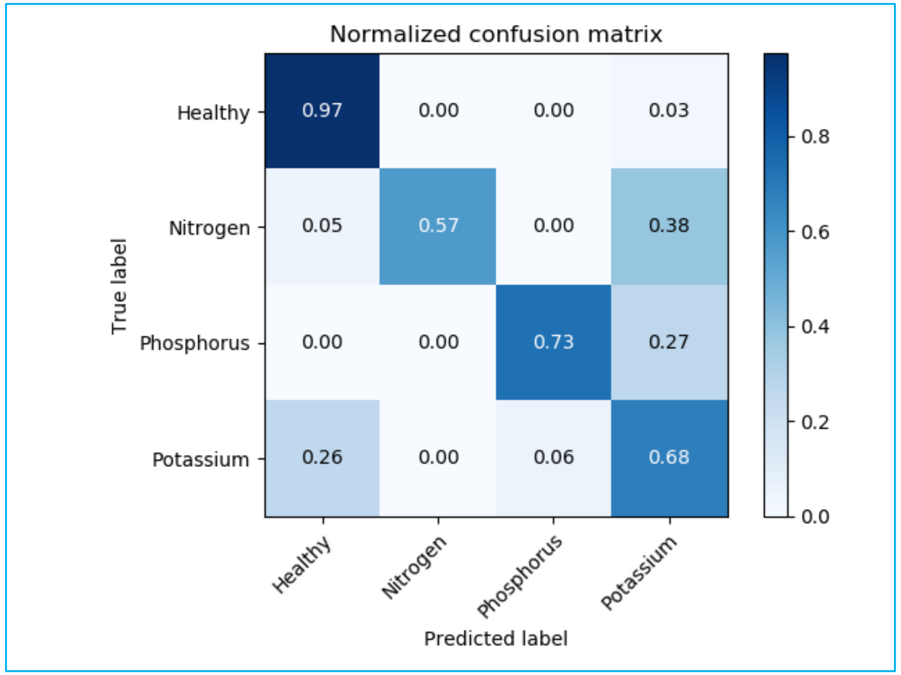
decision_function_shape='ovr' or 'ovo'

Decision function shape → Default
Kernel → Default

Accuracy score: 80.75%
Training Set Mean Absolute Error: 50%
Testing Set Mean Absolute Error: 47%

Accuracy score: 0.8074534161490683

	precision	recall	f1-score
0	0.84	0.97	0.90
1	1.00	0.57	0.73
2	0.79	0.73	0.76
3	0.71	0.68	0.69
accuracy			0.81
macro avg	0.83	0.74	0.77
weighted avg	0.81	0.81	0.80

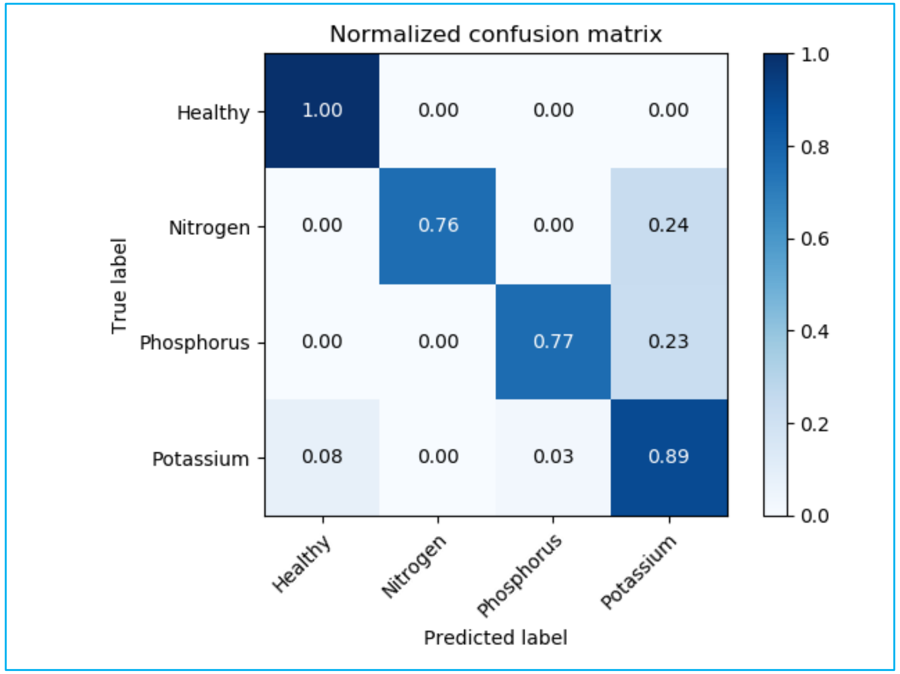


Decision function shape → 'OvR'
Kernel → 'Linear'
C → 1000

Accuracy score: 91.47%
Training Set Mean Absolute Error: 13%
Testing Set Mean Absolute Error: 16%

Accuracy score: 0.9147286821705426

	precision	recall	f1-score
0	0.95	1.00	0.98
1	1.00	0.76	0.87
2	0.91	0.77	0.83
3	0.83	0.89	0.86
accuracy			0.91
macro avg	0.92	0.86	0.88
weighted avg	0.92	0.91	0.91



Test and Results (Healthy leaves)

Class	Color Space	Lower boundaries	Upper boundaries	Contour Area
Healthy	HSV	[33,39,61]	[74,255,255]	Greater than 500

Test Images						
Total Detected	19	24	22	14	15	12
False Detected	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	5 (42%)
Classified Correctly	15 (79%)	23 (96%)	20 (91%)	14 (100%)	15 (100%)	3 (25%)
Misclassified	4 (21%)	1 (4%)	2 (9%)	0 (0%)	0 (0%)	4 (33%)

- “total detected” – number of leaf images detected from the plant image with different backgrounds
- “false detected” - number of non-leave images detected
- “classified correctly” - number of correctly classified images for appropriate classes
- “misclassified” - number of incorrect classified images as other classes.

Test and Results (Nitrogen Deficiency leaves)

Class	Color Space	Lower boundaries	Upper boundaries	Contour Area
Nitrogen	HSV	[0,173,119]	[37,255,255]	Greater than 500

Test Images						
Total Detected	1	2	4	6	7	6
False Detected	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Classified Correctly	1 (100%)	1 (50%)	4 (100%)	5 (83%)	7 (100%)	3 (50%)
Misclassified	0 (0%)	1 (50%)	0 (0%)	1 (17%)	0 (0%)	3 (50%)

Test and Results (Phosphorus Deficiency leaves)

Class	Color Space	Lower boundaries	Upper boundaries	Contour Area
Phosphorus	RGB	[0,0,157]	[146,137,255]	Greater than 500

Test Images						
Total Detected	3	1	3	0	3	3
False Detected	0 (0%)	0 (0%)	0 (0%)	-	0 (0%)	0 (0%)
Classified Correctly	3 (100%)	1 (100%)	3 (100%)	-	3 (100%)	3 (100%)
Misclassified	0 (0%)	0 (0%)	0 (0%)	-	0 (0%)	0 (0%)

Test and Results (Potassium Deficiency leaves)

Class	Color Space	Lower boundaries	Upper boundaries	Contour Area
Potassium	HSV	[0,0,53]	[57,255,255]	Greater than 500

Test Images						
Total Detected	19	18	14	9	10	30
False Detected	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	4 (13%)
Classified Correctly	3 (16%)	3 (17%)	6 (43%)	3 (33%)	5 (50%)	7 (23%)
Misclassified	16 (84%)	15 (83%)	8 (57%)	6 (67%)	5 (50%)	19 (64%)

Nutrient solutions for Strawberry Plants [5]

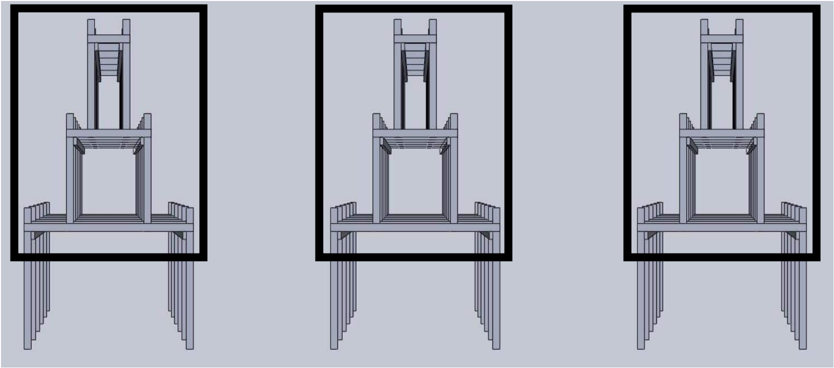
Nutrient	Main functions	Fertilizers	Application Rate (L/1000 Plants)
Nitrogen(N)	Growth and yield	Urea (CH_4N_2O) <46%N>	0.4 – 0.5
Phosphorus(P)	Fruit development	Triple Superphosphate ($CaH_4P_2O_8$) <46%P>	2.5 – 3
Potassium (K)	Fruit quality and flavor	Potassium Chloride (KCl) <60%K>	0.7 – 0.8

- NPK should be applied carefully to get better yield and fruit quality.

Soluble in water

Fertigation (Fertilizers + Irrigation)

Fertilizers	Application Rate (liter/55 Plants)	Drip Pipe Flow Rate	Supply Duration (minutes/55 plants/2 weeks)
Urea	0.022 – 0.0275	1.95 liters per hour	1
Triple Superphosphate	0.1367 – 0.165		5
Potassium Chloride	0.0385 – 0.044		2



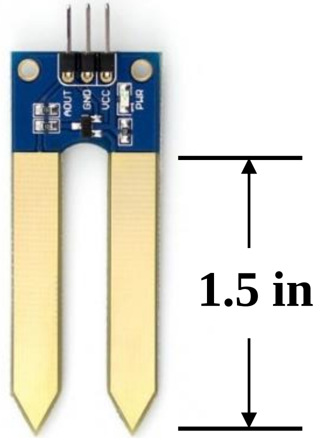
- There are three stands in the farm and each stand has 55 plants for five rows.

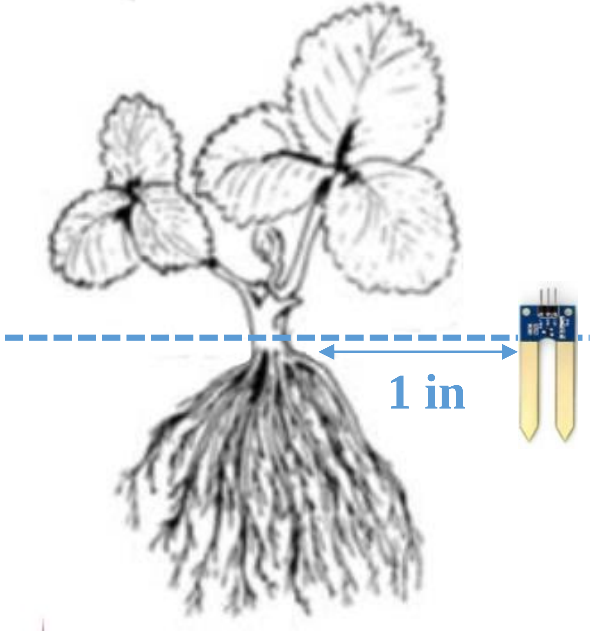
Drip irrigation System

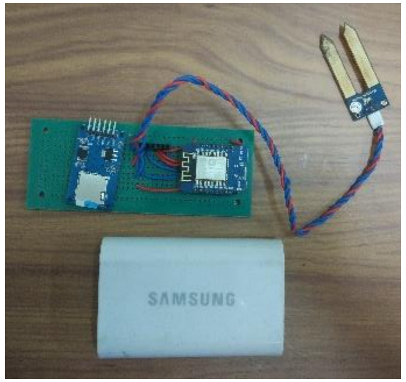
Processes	Requirements
Soil Moisture Data Logging	Waveshare Soil Moisture Sensor, ESP8266 D1 Mini Board, SD card, SD card Module, Power bank
Comparison and Scheduling for Water Supply Duration	Soil Moisture Data from Different Farms

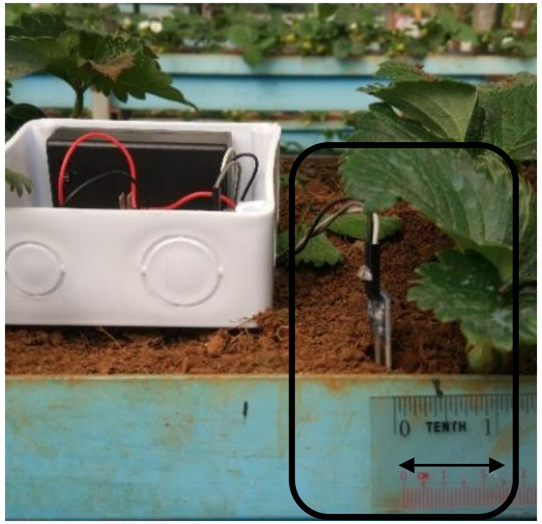
Soil Moisture Data Logging

Waveshare Soil Moisture Sensor









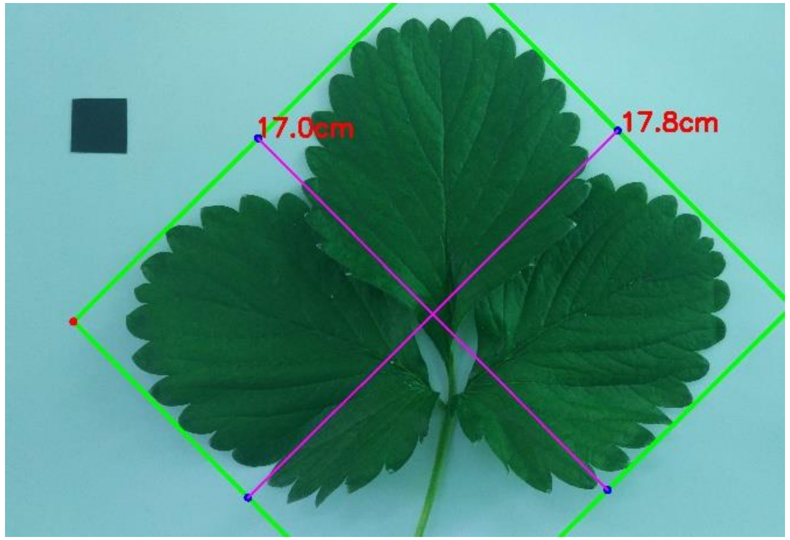
- moisture values are required to set to avoid over-watered or under-watered.
- too little moisture ➔ plant death.
- too much ➔ root disease and wasted water.

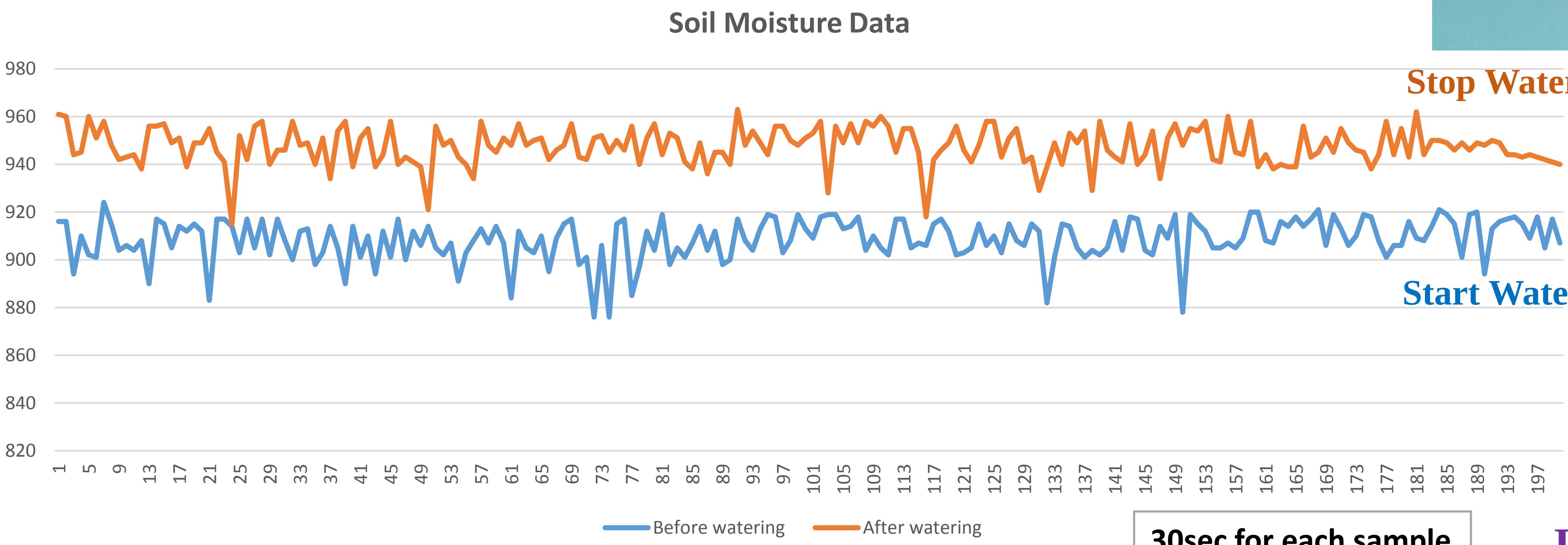
Strawberry Farms around POL

Fields	Aung Chan Thar	City Farm-A	City Farm-B	City Farm-C
Growing System	Indoor	Outdoor	Outdoor	Indoor
Farm Structure				
Supply Method	Drip System	Drip System	Drip System	Drip System
Supply Duration	20 minutes (twice a day)	15 minutes (twice a day)	15 minutes (twice a day)	15 minutes (thrice a week)

Soil Moisture Data for Small-scale Farm

- **Life** - Two months old plants
- **Water supply Duration** - 10 minutes (twice a day)
- **Data logging Time** - Before watering (10 min)
  - After watering (10 min)





Stop Watering (948)

Start Watering (908)

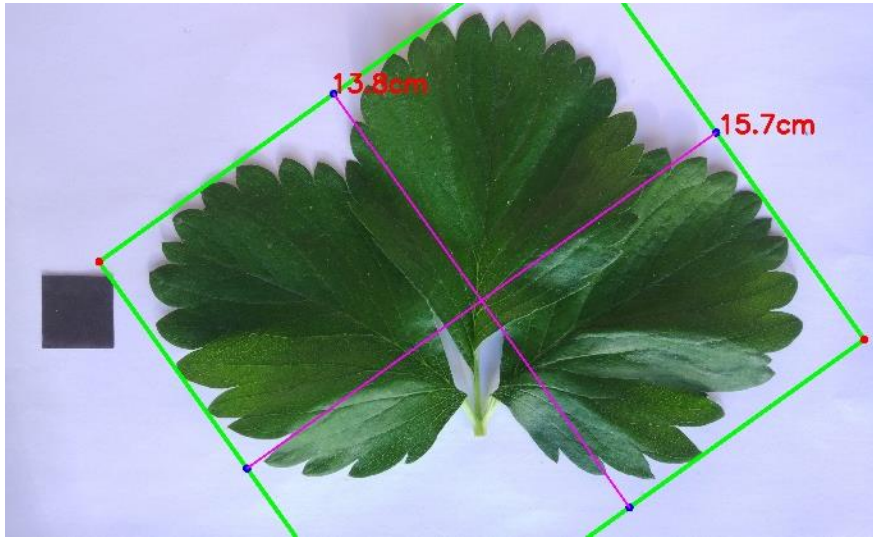
30sec for each sample

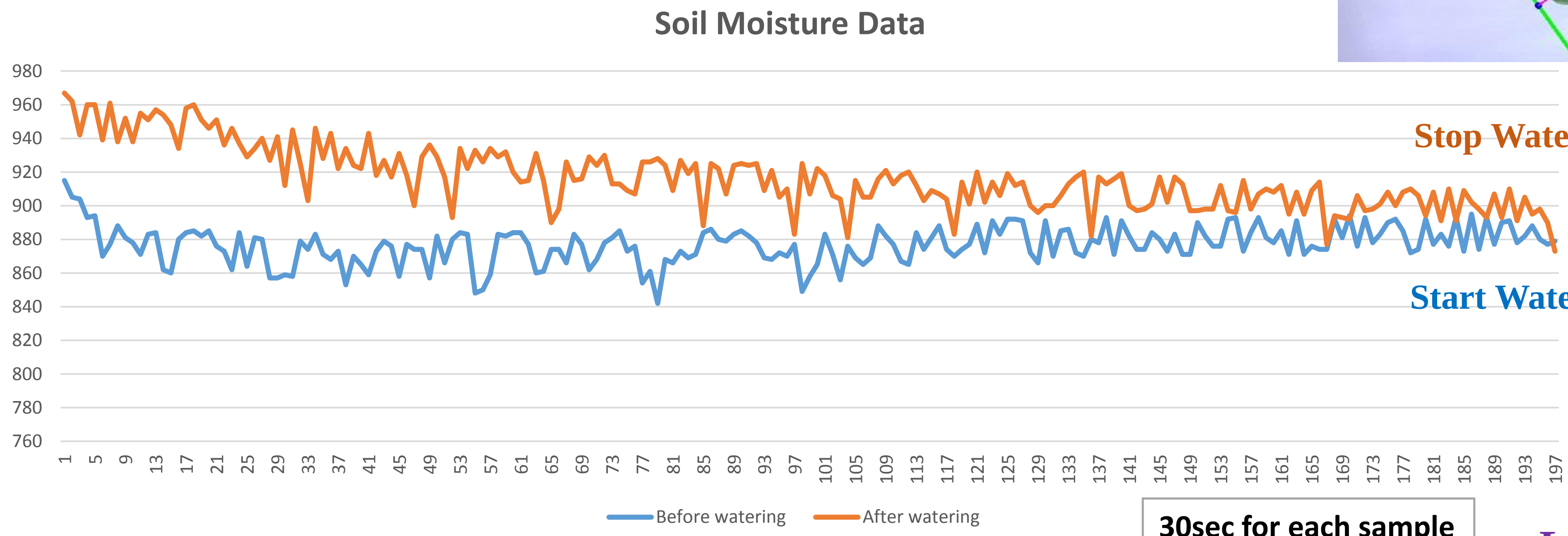
Moisture : (908 – 948)

Leaf size : 17.0cm x 17.8cm

Soil Moisture Data for Aung-Chan-Thar

- **Life** - Two months old plants
- **Water supply Duration** - 20 minutes (twice a day)
- **Data logging Time** - Before watering (10 min)
  - After watering (10 min)





Stop Watering (918)

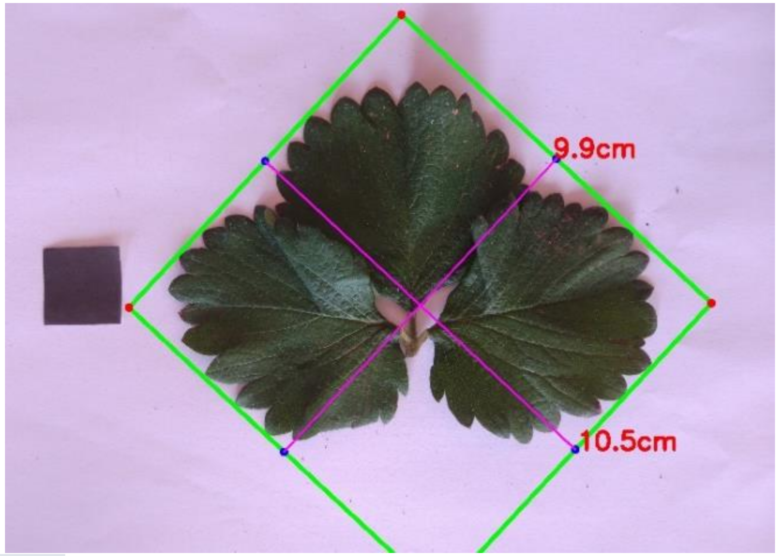
Start Watering (876)

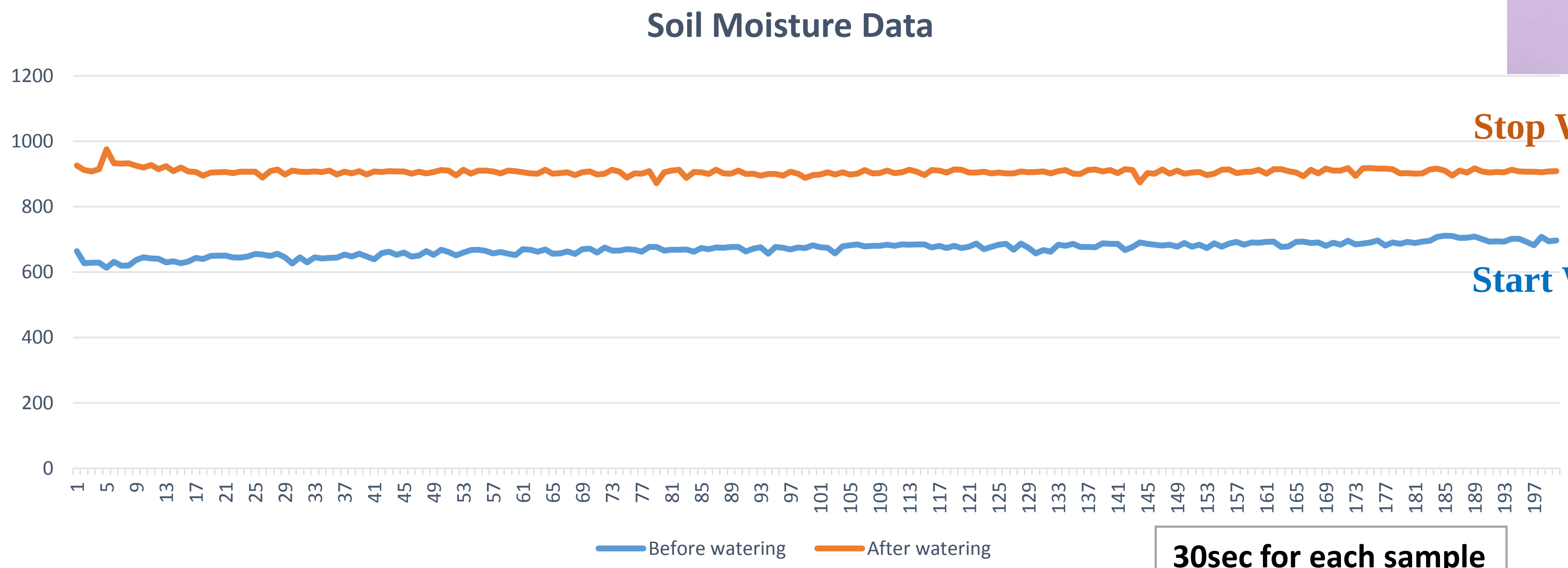
30sec for each sample

Moisture : (876 – 918)
Leaf size – 13.8cm x 15.7cm

Soil Moisture Data for City Farm-A

- **Life** - Two months old plants
- **Water supply Duration** - 15 minutes (twice a day)
- **Data logging time** - Before watering (10 min)
  - After watering (10 min)





Stop Watering (907)

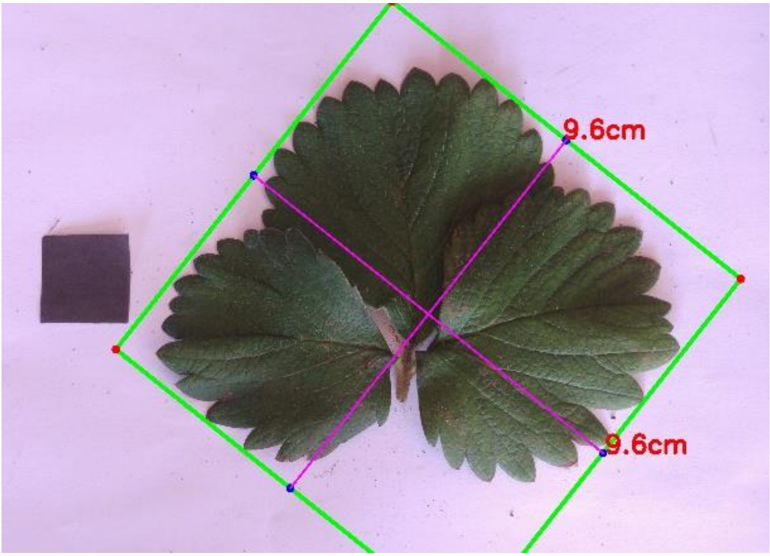
Start Watering (670)

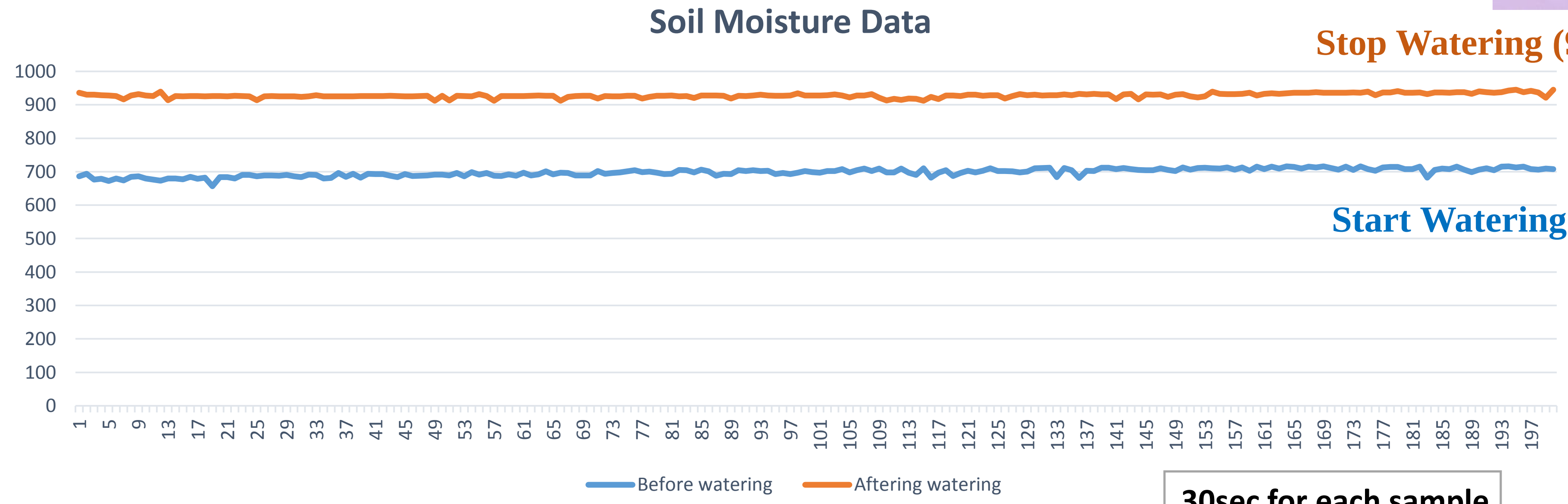
30sec for each sample

Moisture : (670 – 907)
Leaf size – 10.5cm x 9.9cm

Soil Moisture Data for City Farm-B

- **Life** - Two months old plants
- **Water supply Duration** - 15 minutes (twice a day)
- **Data logging time** - Before watering (10 min)
  - After watering (10 min)





Stop Watering (928)

Start Watering (697)

30sec for each sample

Moisture : (697 – 928)

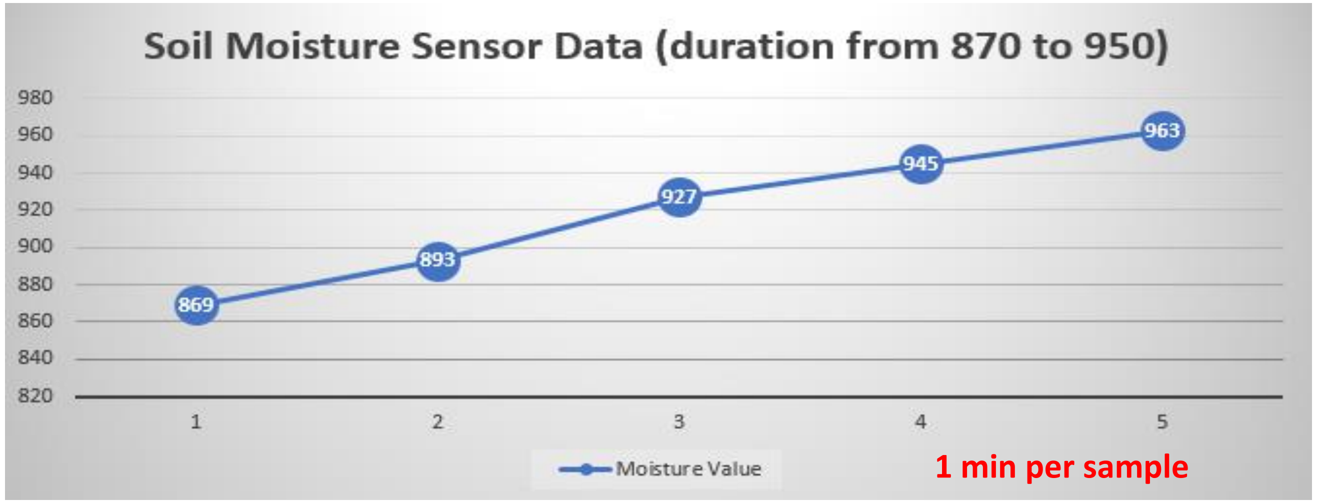
Leaf size – 9.6cm x 9.6cm

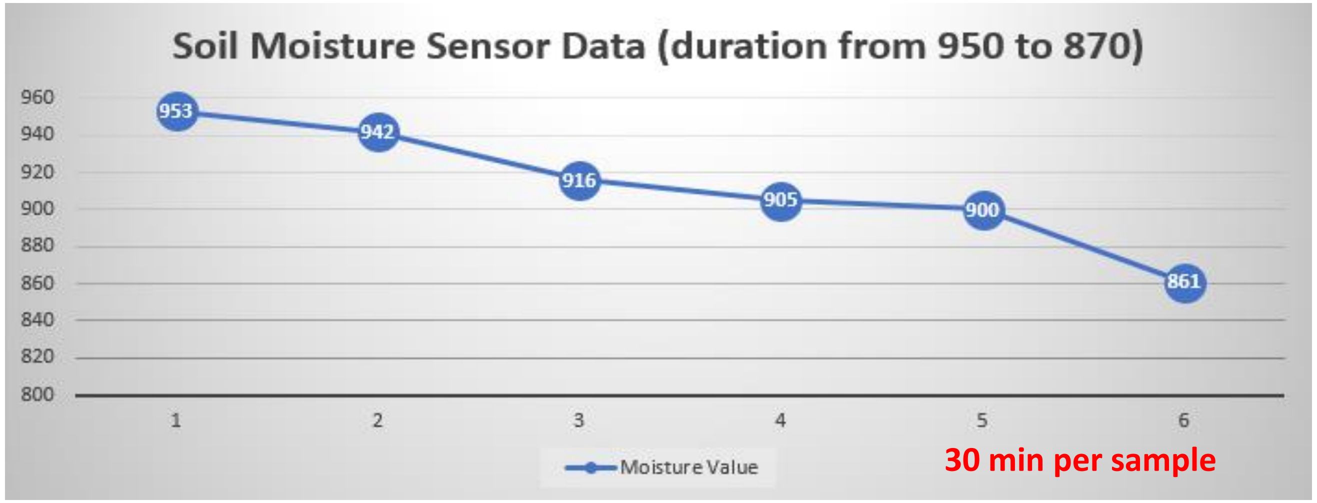
Comparison of Soil Moisture Data and Leaf Size

Fields	Growing System	Moisture		Leaf Size
		Min	Max	
Small-scale Farm	Indoor	908	948	17.0cm x 17.8cm ✓
Aung Chan Thar	Indoor	876	918	13.8cm x 15.7cm ✓
City Farm-A	Outdoor	670	907	10.5cm x 9.9cm
City Farm-B	Outdoor	697	928	9.6cm x 9.6cm
City Farm-C	Indoor	720	927	13.3cm x 13.7cm ✓
Optimal State	**Indoor**	**870**	**950**	

❖ **The larger the leaf area, the more production rate we can get.**

Scheduling for Water Supply Duration





Minimum – 870
Maximum – 950

Moisture Range	Duration
870 to 950 (watering)	5 minutes
950 to 870 (normal)	3 hours

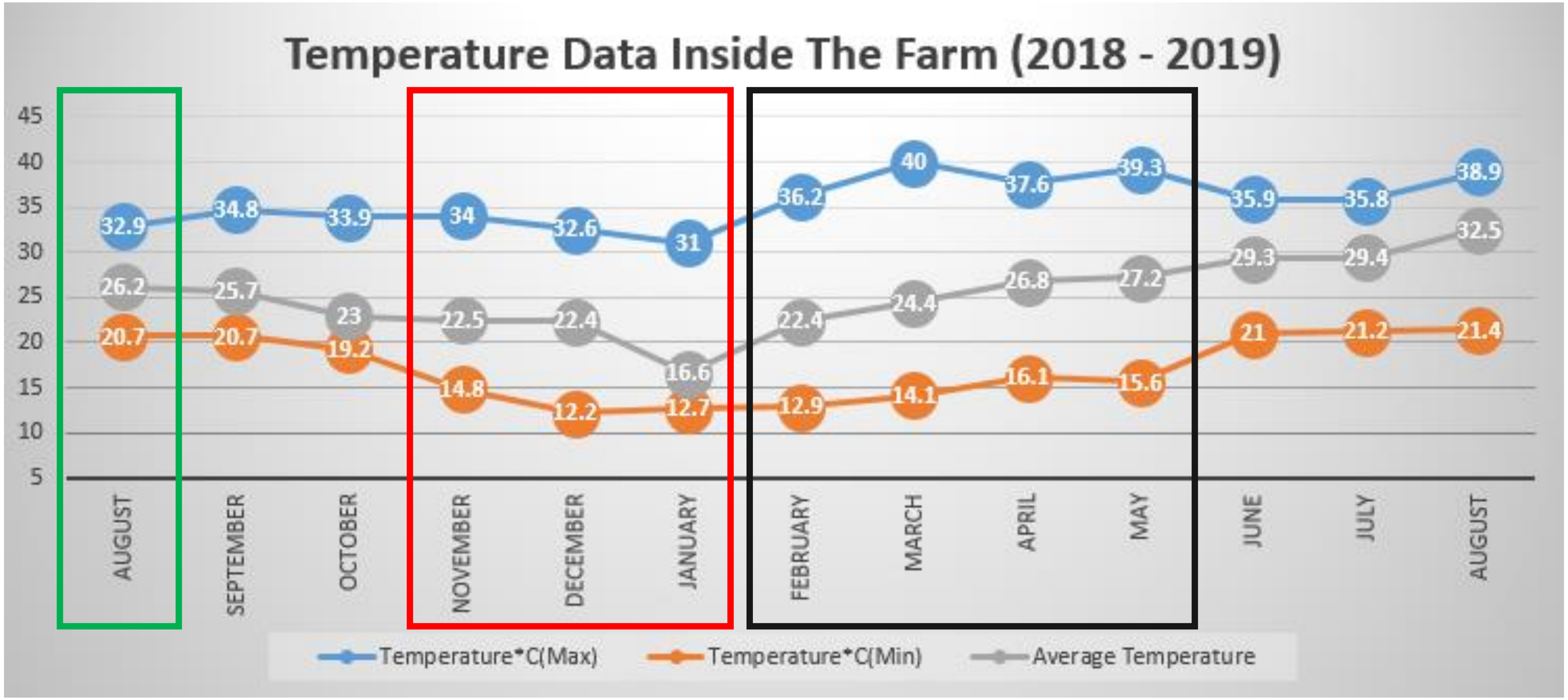
Schedule		Duration	Drip Pipe Flow Rate	Water Usage
3 AM	3 PM	5 min	1.95 liters per hour	1.3 liter/plant/day
6 AM	6 PM	5 min		
9 AM	9 PM	5 min		
12 AM	12 PM	5 min		

72 liters (19 gallons) per day for all 55 plants

Temperature Control System

Processes	Requirements
Temperature Data Logging	DHT22, Raspberry Pi 3, Real Time Clock, Memory Stick
Temperature Controlling	Exhaust Fan, Sprinklers, Cooling Pad
Scheduling for Auto Temperature Control System	Logged Temperature Data for Each Season

Inside Temperature from August,2018 to May,2019

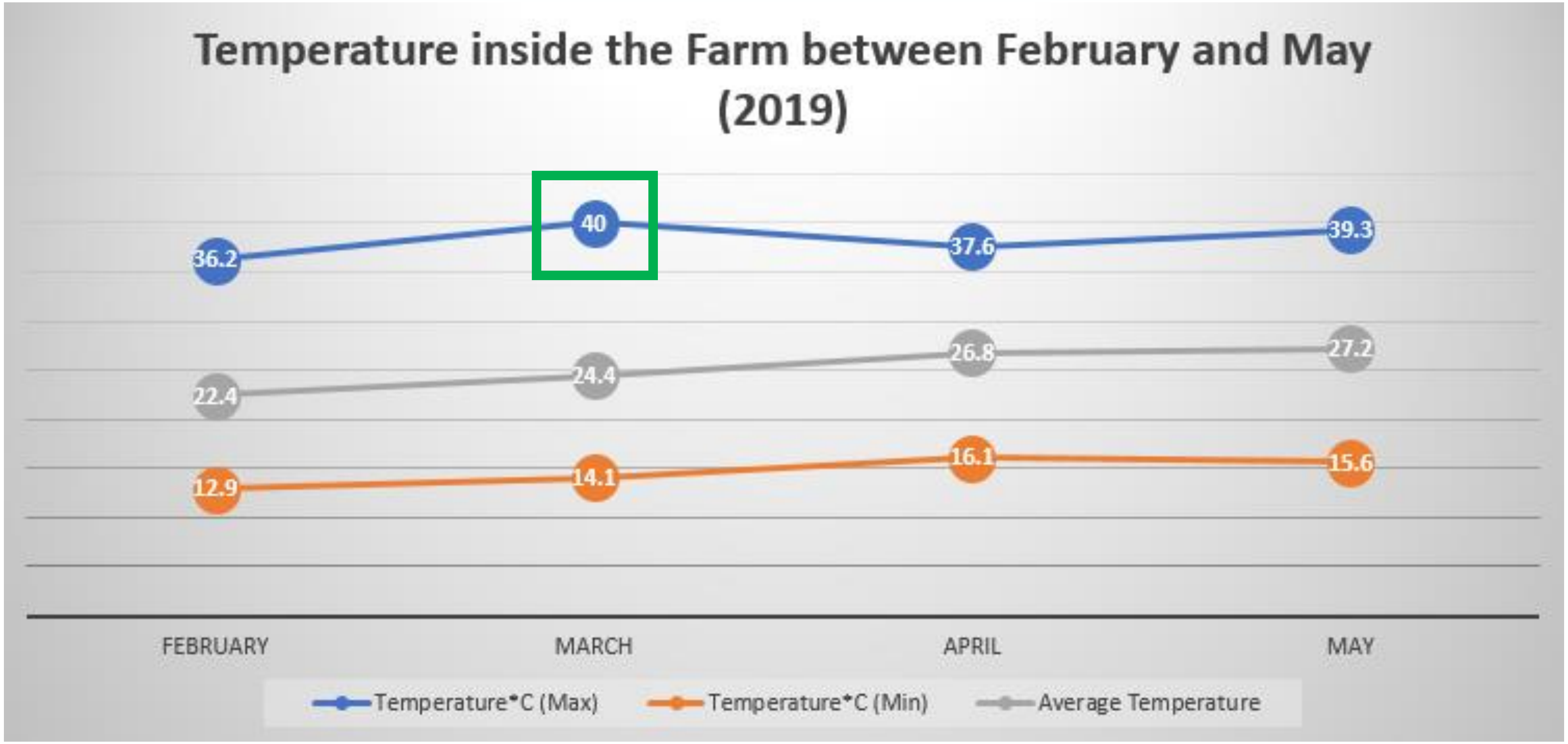


Green box – started fruiting

Red box – plenty of fruits

Black box – a little amount of fruits because of high temperature

Temperature for hot season



(20°C ~ 29°C) ⟶ Set fruits

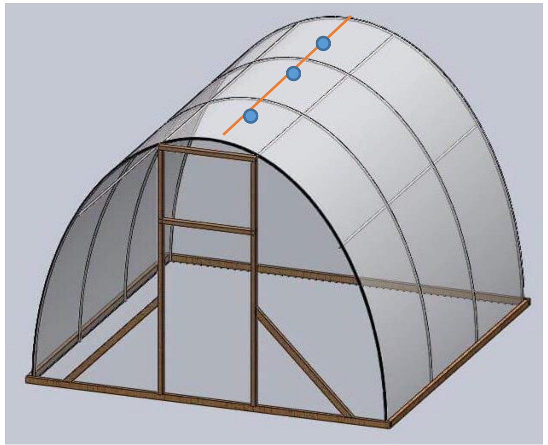
Less than 10 °C ⟶ Fail to germinate

Need to reduce 11 degree Celsius

How to control temperature inside the farm?

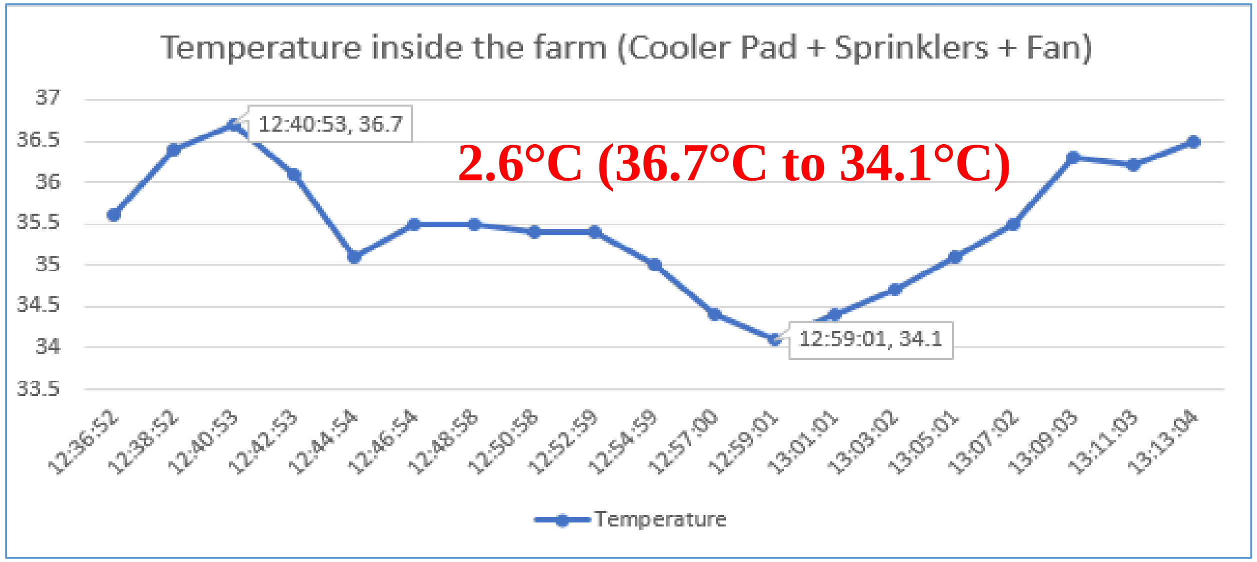
Exhaust Fan

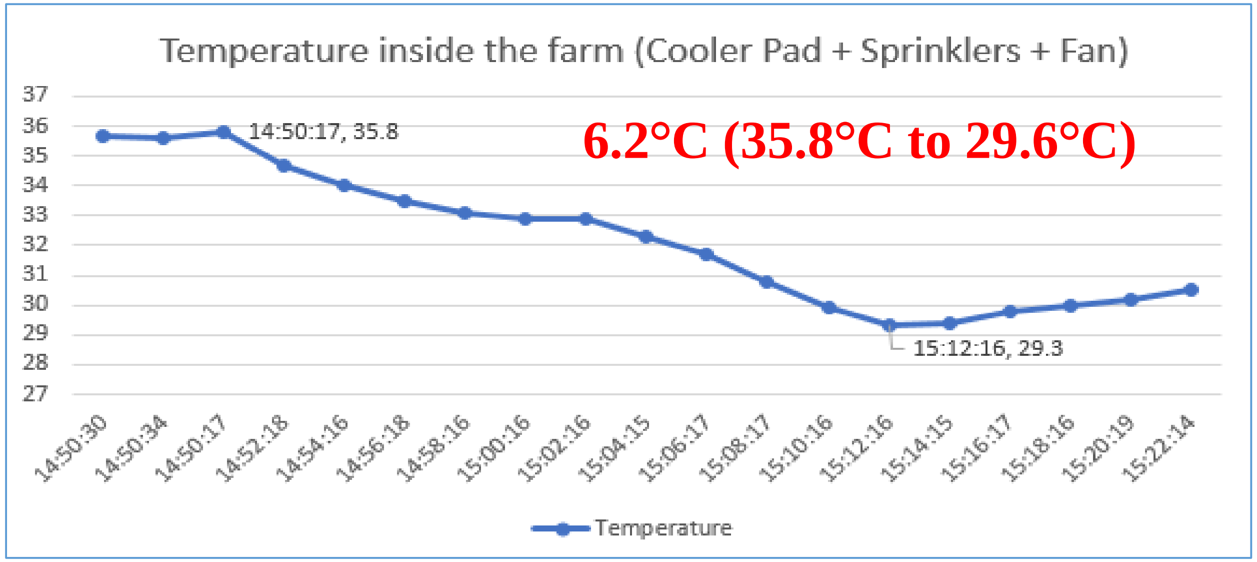
Sprinklers

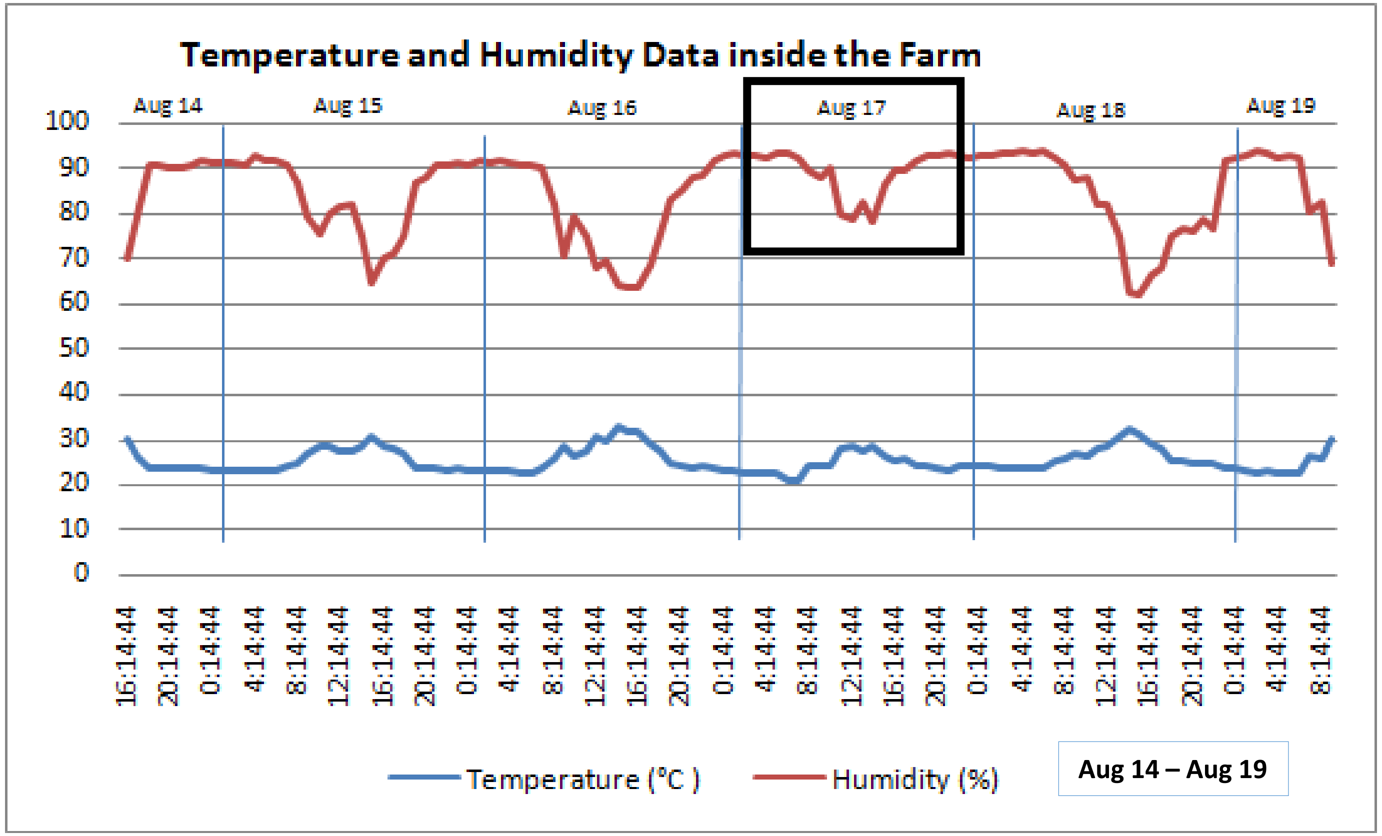


Cooler Pad

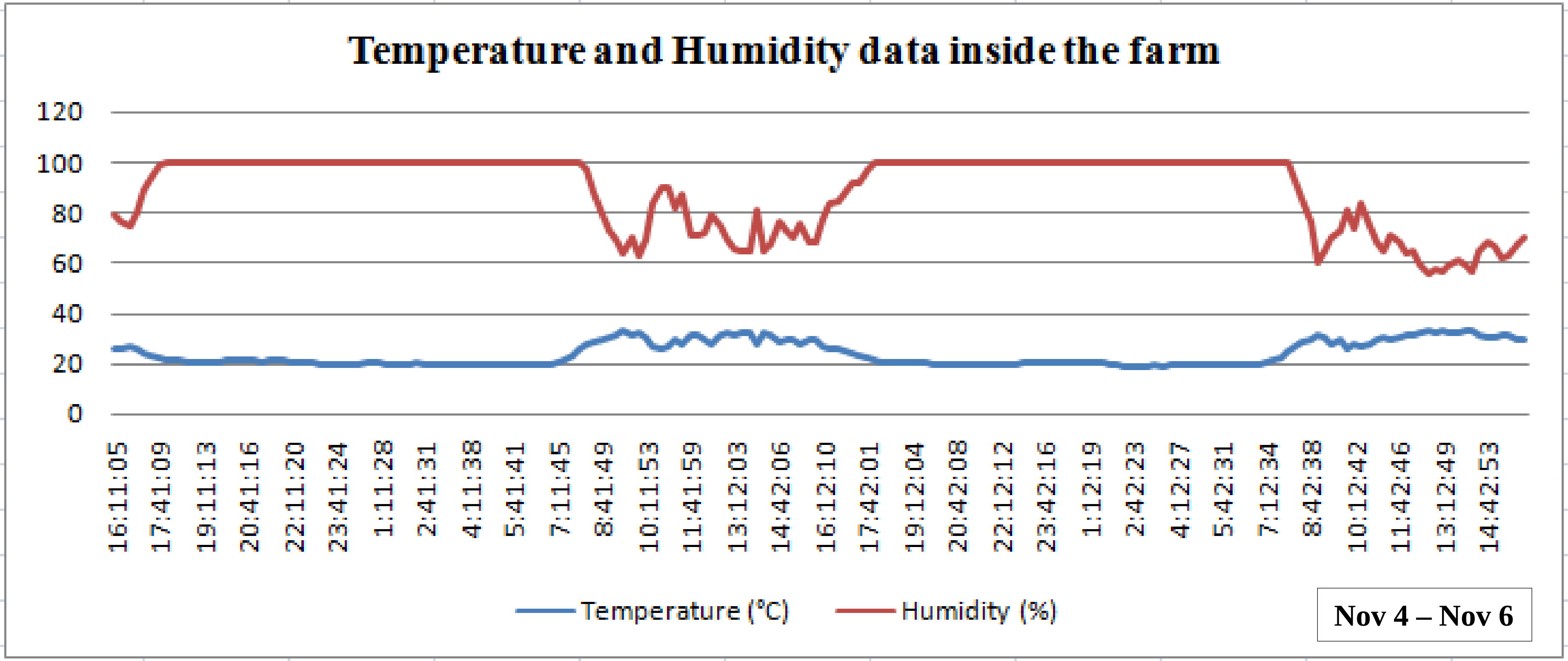
Temperature control inside the farm?







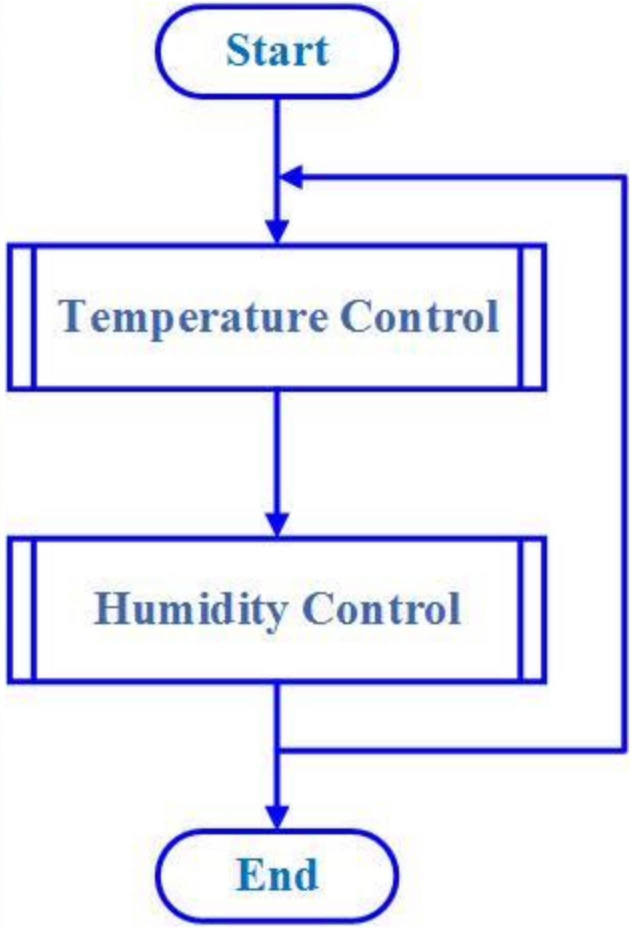
- data logged during rainy season.
- humidity is high (average of 93%) , leaf scorch and bacterial blight are affected.
- need to enable good aeration for plants.



- data logged at the start of cold season.

- Temperature is in the range (19°C to 31°C), with the average of 23°C.
- Average humidity is 89%.

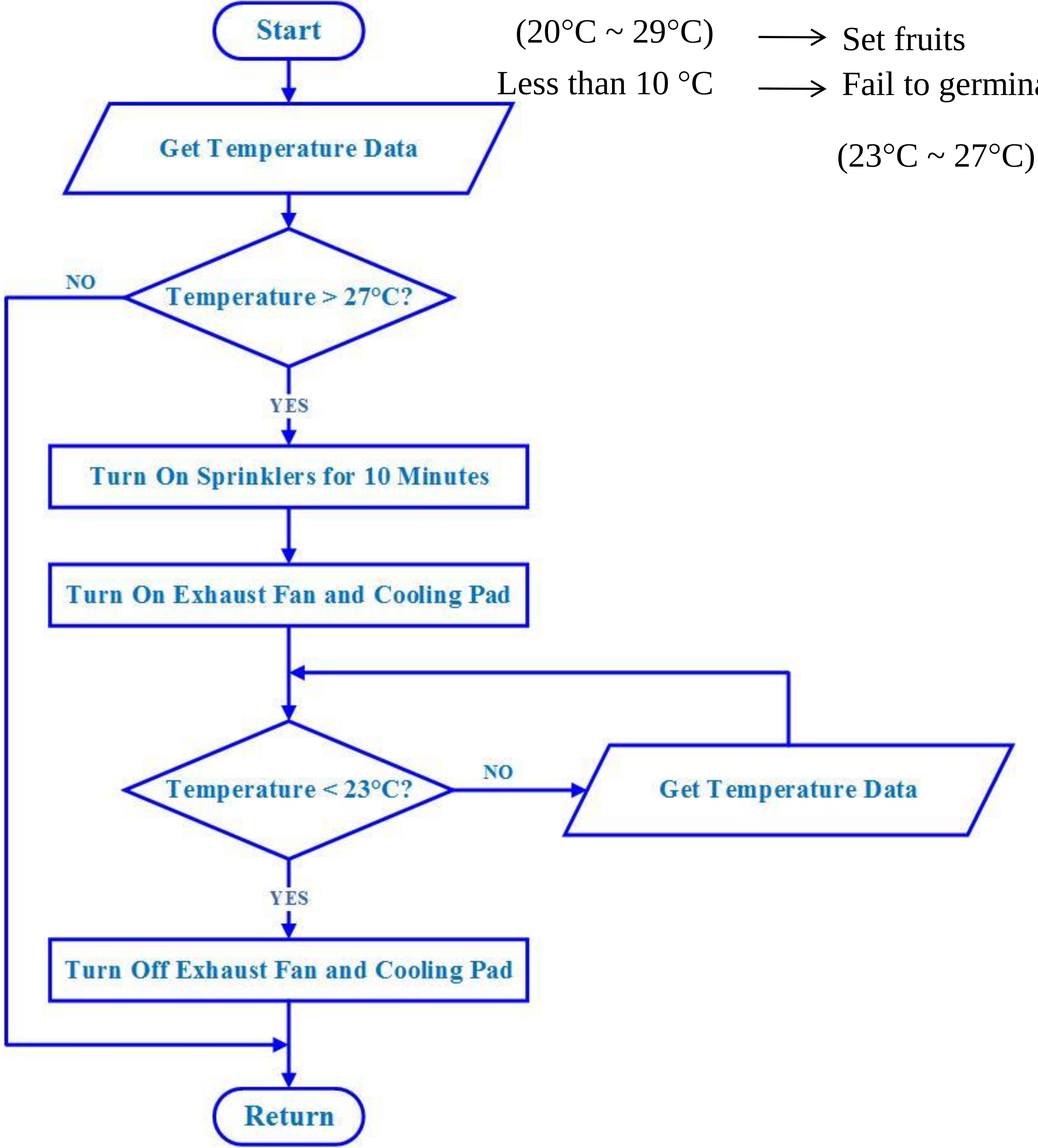
System Flowchart



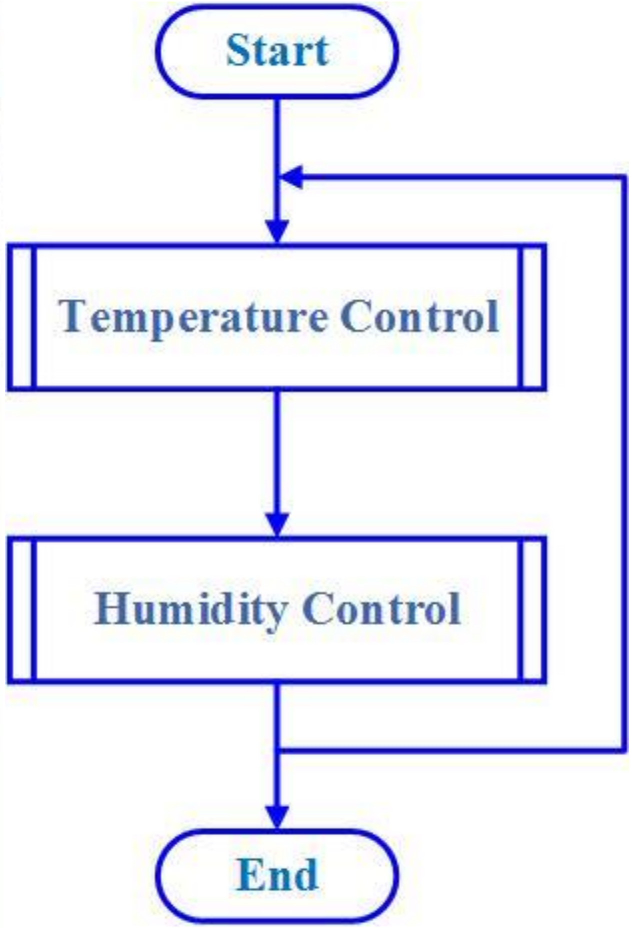
(20°C ~ 29°C) → Set fruits

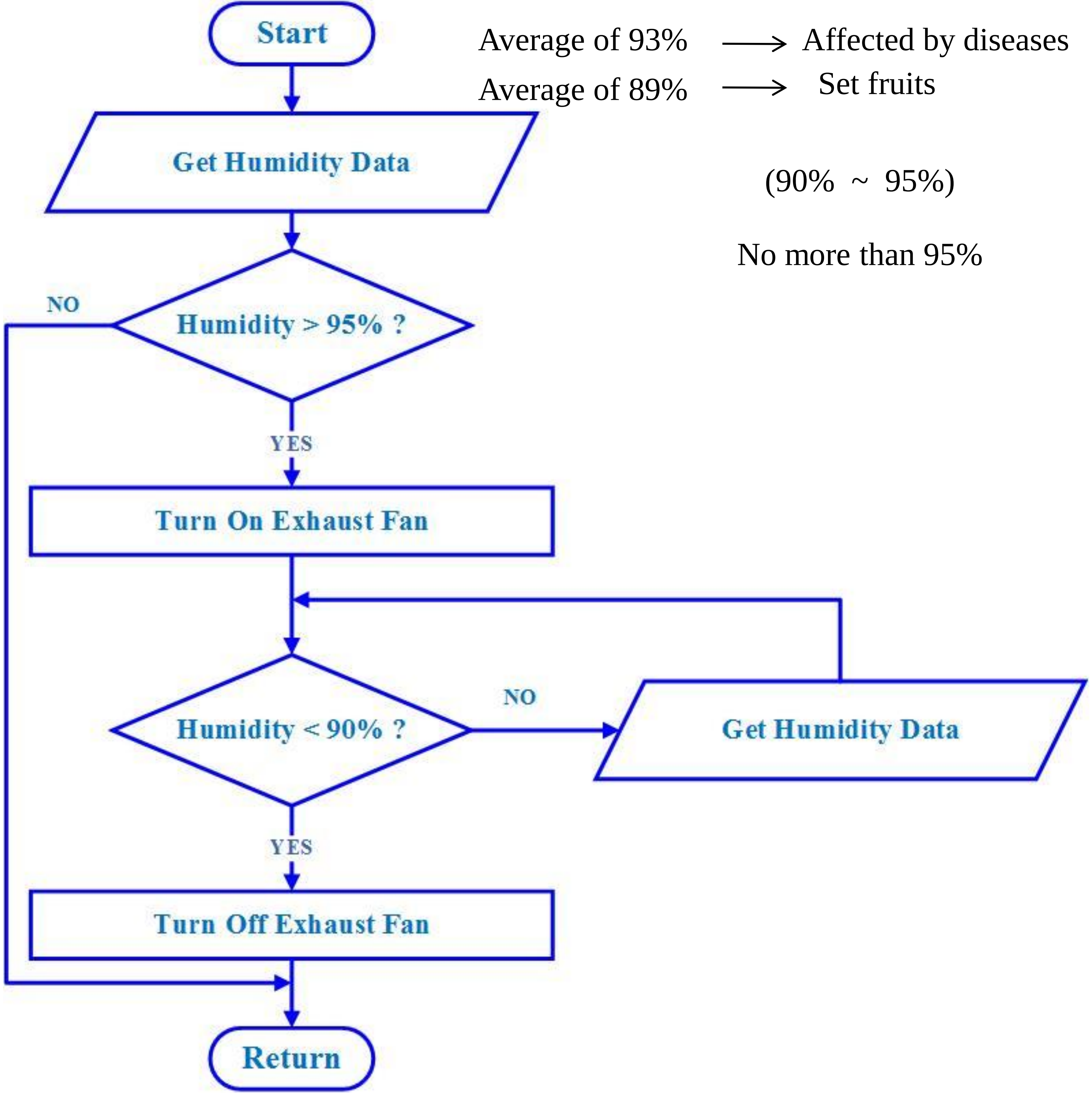
Less than 10 °C → Fail to germinate

(23°C ~ 27°C)



System Flowchart





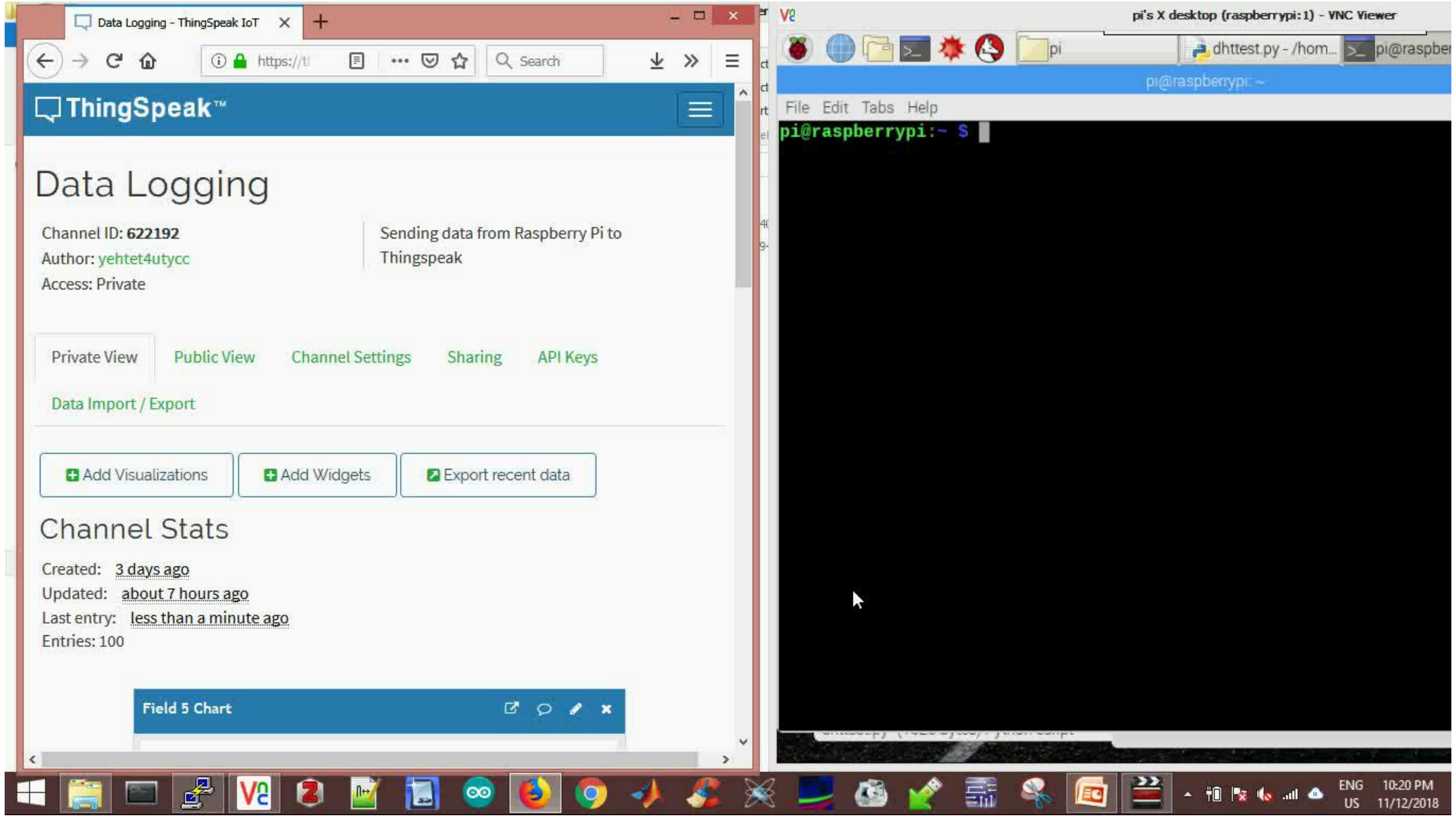
Average of 93% ⟶ Affected by diseases

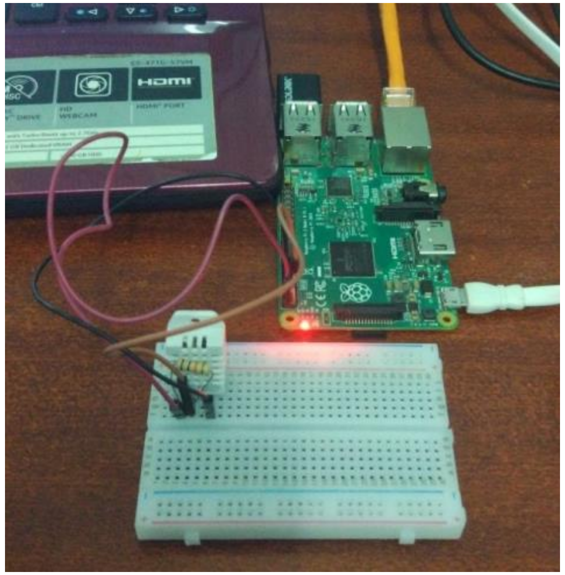
Average of 89% ⟶ Set fruits

(90% ~ 95%)

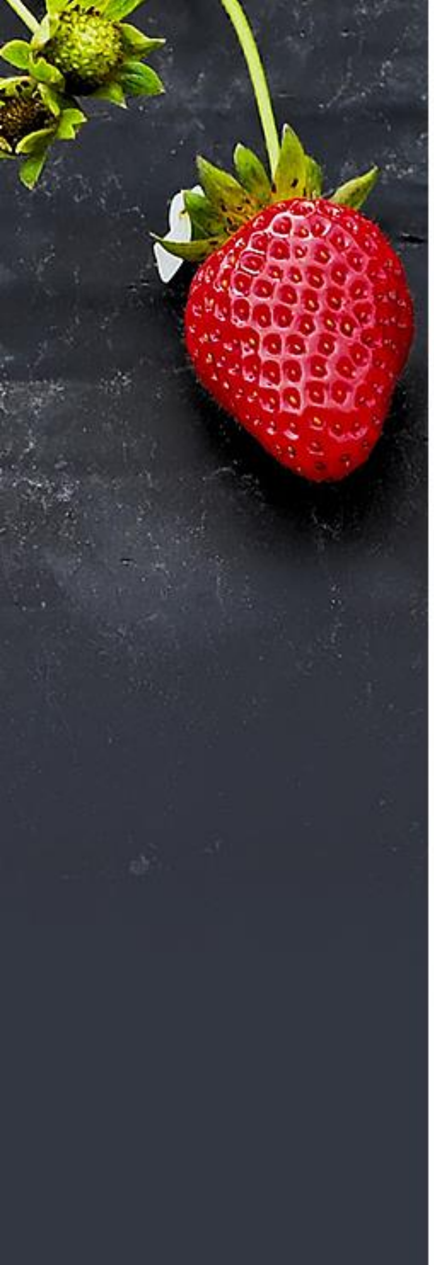
No more than 95%

Updating Data to ThingSpeak





- Thingspeak takes at least 15 seconds to update data.
- sampling time is set 20 seconds for each.



GUI design

<u>Why is UI design important?</u>

- User interfaces allows end users to interact with application.
- A good UI will make an application intuitive and easy to use.

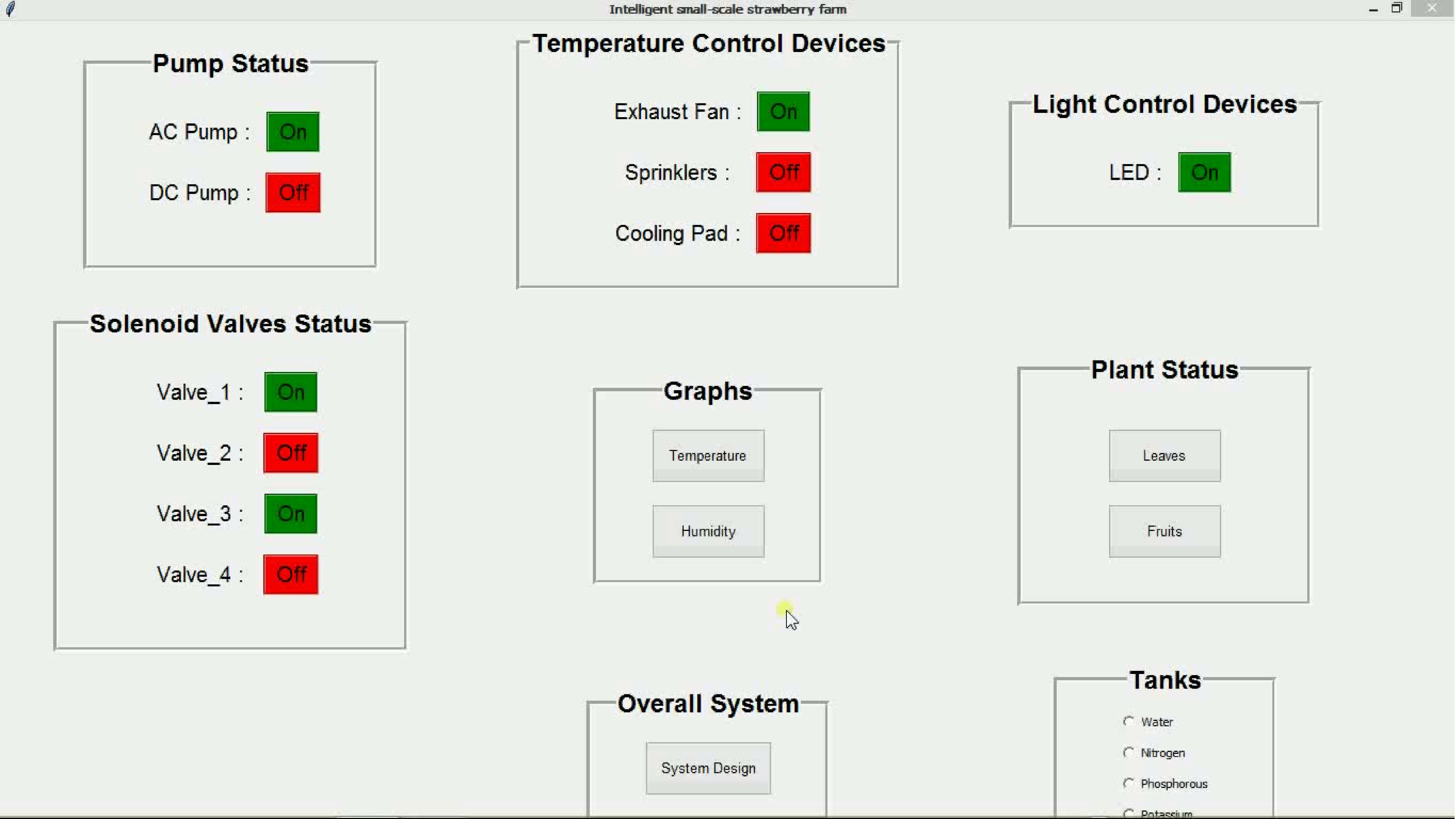
Tkinter **wxPython** **JPython (Jython)** **PyKDE / PyQt** **PyGTK**

X11 **WPY** **Win32all.exe**

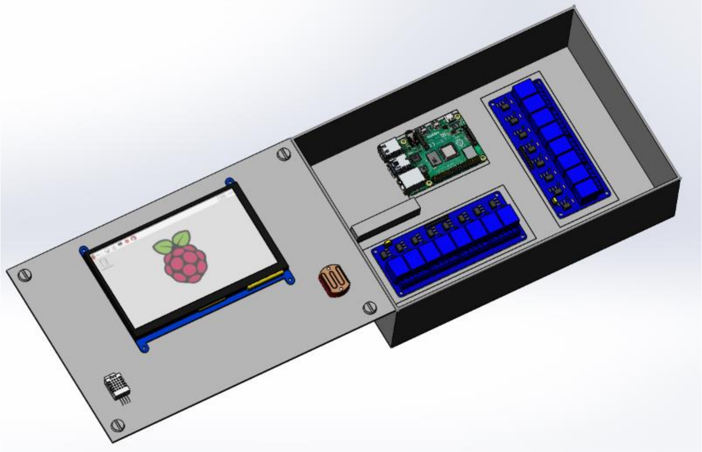
Tkinter

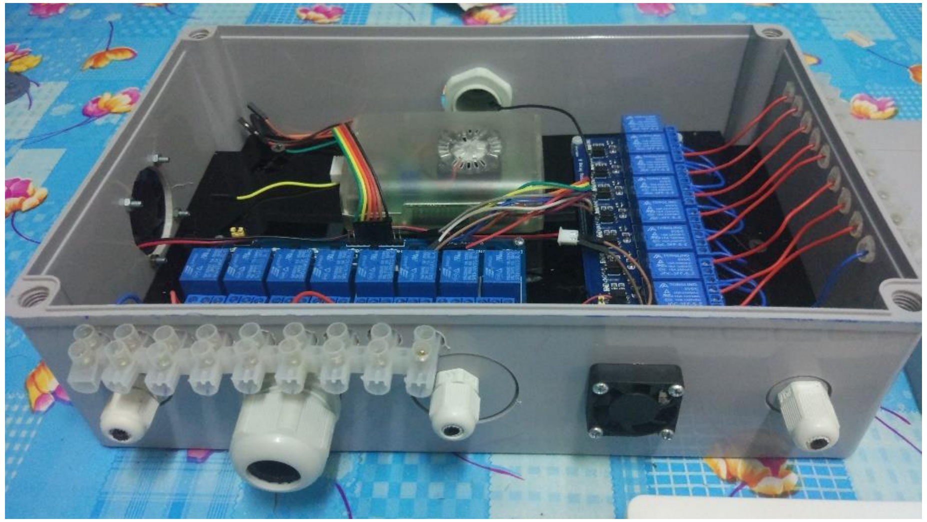
The Tkinter module (“Tk interface”) is the standard Python interface to the Tk GUI toolkit [7].

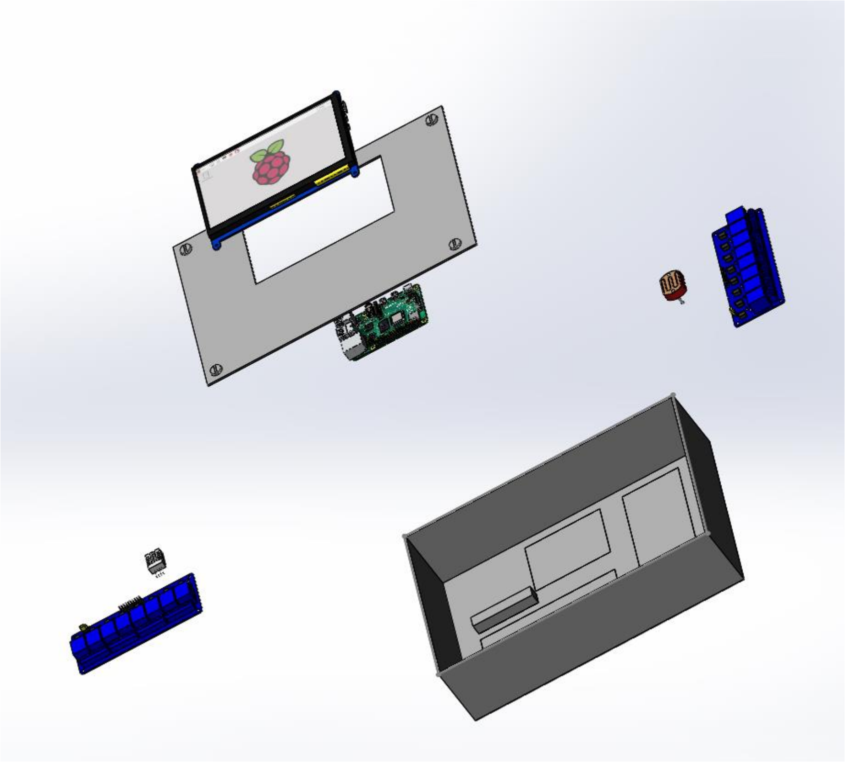
layered design, accessibility learning, portability, and availability.

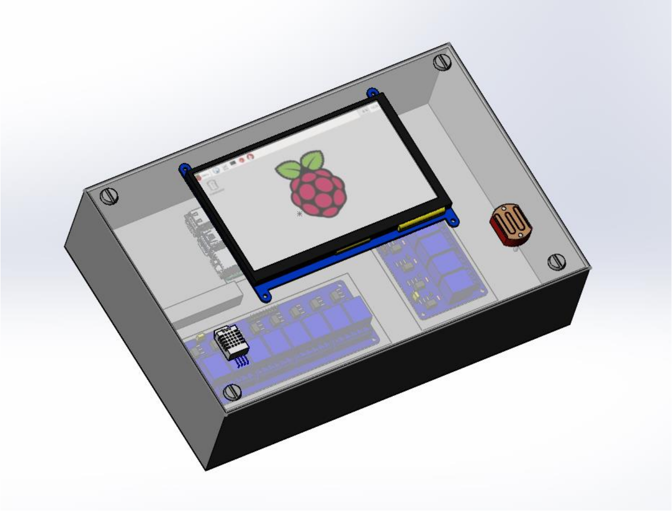

Intelligent small-scale strawberry farm
Pump Status
AC Pump :
On
DC Pump :
Off
Temperature Control Devices
Exhaust Fan :
On
Sprinklers :
Off
Cooling Pad :
Off
Light Control Devices
LED :
On
Solenoid Valves Status
Valve_1 :
On
Valve_2 :
Off
Valve_3 :
On
Valve_4 :
Off
Graphs
Temperature
Humidity
Plant Status
Leaves
Fruits
Overall System
System Design
Tanks
Water
Nitrogen
Phosphorous

Control Panel





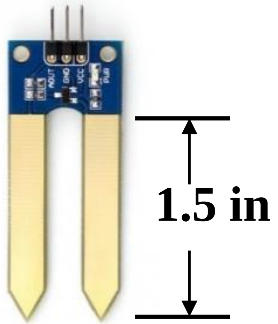


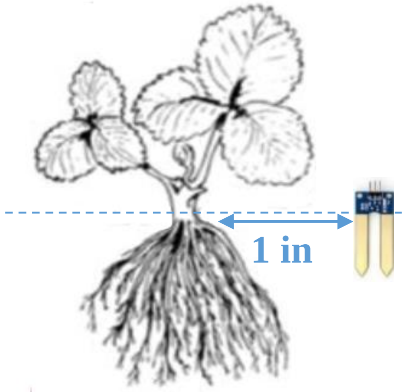


Discussion

- Soil moisture sensor’s length is only 1.5 inches and cannot sense the root zones deeply.
- Data are recorded from just four farms in Pyin Oo Lwin and the plants may have different soil types, soil mixing ingredients, environmental temperatures, water and fertilizer supply.
- 15 solenoid valves should be used for all 15 plant rows to supply optimal water and fertilizers to only appropriate rows.
- Further testing on the feature extraction and classification algorithm could also be the future tasks.

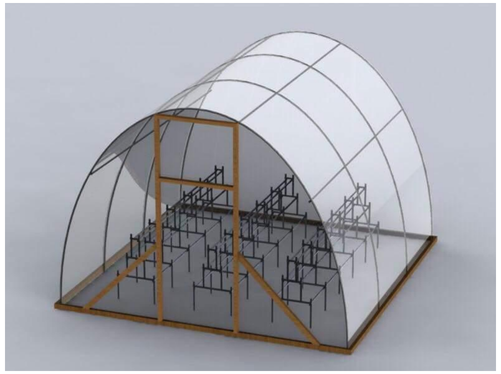




Discussion (Cont'd)

- Much amount of temperature (at least 11°C) inside the farm is needed to reduce during the hot season in order to set fruits.
- Using a high-resolution camera with motorized camera slider will reduce the cost for multiple camera usage and also increase the robustness.
- Plants in the middle stand have better effect than other two stands.
    - The leaves are large, more flowers are set and the fruits are bigger.
    - Not directly touched by sunlight (can stay in the shadow).



Conclusion

- Pyin Oo Lwin is the suitable place for strawberry plants but fruits bear once a year.
- Plants can reflect weather and soil conditions.
- Indoor system is more efficient than traditional open fields.
- Drip irrigation saves water and fertilizer ensuring optimal growth in low cost, high reliability and accuracy.
- Implementation of leaf analysis can effectively monitor plant growth trends and detect nutrient deficiency symptoms.

References

[1] C. Joseph, I. Thirunavuakkarasu, A. Bhaskar, and A. Penujuru, “Automated fertigation system for efficient utilization of fertilizer and water,” 2017, pp. 1–6.

[2] I. Mohanraj, V. Gokul, R. Ezhilarasie, and A. Umamakeswari, “Intelligent drip irrigation and fertigation using wireless sensor networks,” 2017, pp. 36–41.

[3] V. Pooja, R. Das, and V. Kanchana, “Identification of plant leaf diseases using image processing techniques,” 2017, pp. 130–133.

[4] M. V. Latte and S. Shidnal, “Multiple nutrient deficiency detection in paddy leaf images using color and pattern analysis,” 2016, pp. 1247–1250.

[5] “Nutritional Recommendations for Strawberry”. Retrieved from http://www.haifa-group.com/

[6] “Support Vector Machine”. Retrieved from https://scikit-learn.org/stable/modules/svm.html#svm/

[7] B. Dufour, “An Introduction to Python Programming and GUI Design Using Tkinter,”.






Aphids Disease

THANK YOU FOR YOUR ATTENTION

Questions?