Precision Farming

A B M Abir Mahboob, Charles Aresnal Okere, Chiagoziem Cyriacus Ugoh, Muhammad Subhan Khan (Group 6)

Contents

Abstract: This piece of writing is about documenting our project in the Autonomous B lab and that is "Precision Farming". The idea of precision farming is to make it easier for farmers to be involved in more efficient processes whilst farming. This document describes and explains how precision farming can be carried out, with the help of appropriate modeling and implementation using an algorithm.

1 Motivation and introduction (Muhammad Subhan Khan, Charles Aresnal Okere)

We began this research on precision farming by outlining the most common technology used in precision farming. Many studies have been done on a wide range of subjects. Our topic requires us to conduct studies on a certain issue and to acquire a generative notion about precision farming which allows us to get interested in new ideas and approaches.

The primary goal of precision farming is to assist farmers with the latest technologies for managing production and to give farmers the best economical productions. What is precision farming? Precision farming also known as precision agriculture can be defined as the use of accurate information tools or technologies for efficiency.

Precision farming techniques can increase crop production's economic viability and environmental sustainability, thus farmers should be interested in them. Farmers frequently farm each field as a distinct unit in precision farming today.

Information technology is used in precision farming to divide a field into smaller portions and determine the characteristics of each section. The farmer can then apply production inputs in the exact area and quantity required for the highest possible economic yield. A thorough understanding of precision farming requires familiarity with the instruments and methods that support this cutting-edge approach to agricultural management.

With that being said the scenarios that we implemented all had different approaches to the topic. The main idea and the ultimate goal were to make sure to make an accurate and efficient algorithm for the autonomous system of precision farming. However, before implementation is done it is always important to model the scenarios, and this was done using Tapaal.

2 Modeling Scenarios and Diagrams

In this section, the entire group conducted research on precision farming and formulated a multitude of specifications and scenarios. On the basis of such scenarios, we then created multiple diagrams.

2.1 General Activity Diagram (Muhammad Subhan Khan)

As shown in the activity diagram we have the 3 components working, in sync. the operator opens the command center, and after it has opened it, it will sign in and authorize itself, if the operator is authorized then it will be allowed access to the command features, and the command is then posted, the command will wait in the command queue until it is processed. The drone will then get the command and will do either of the 2 things which could be weed or disease detection, depending on that for instance if we take the option of weed detection then it will detect the weed and send the status on the weed, this command will then be sent through the command center to the operator, hence the work will be finished and then the operator can sign out. Examples of the command center, drone, operator simulations will be discussed below.

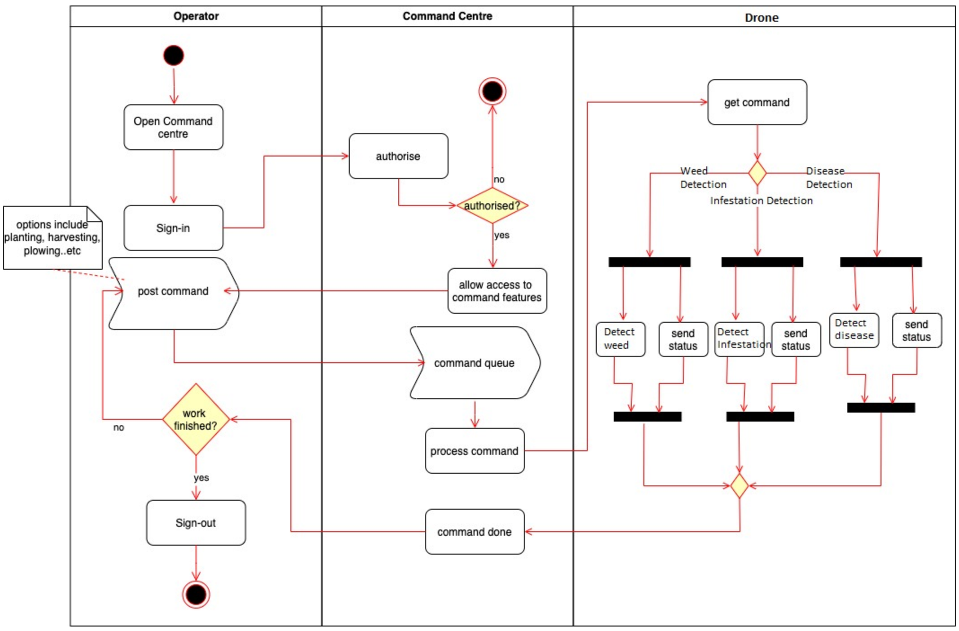


Fig. 1: Activity Diagram

2.2 Infestation Detection (A B M Abir Mahboob)

The monitoring drone will continuously monitor and collect data of the plants, the server will receive this data. This will help the server identify any insect infestation and pinpoint the plants location, specifically the plant which has been infested. The same drone that is monitoring will spray the pesticide as well, after it has accurately navigated the plant.

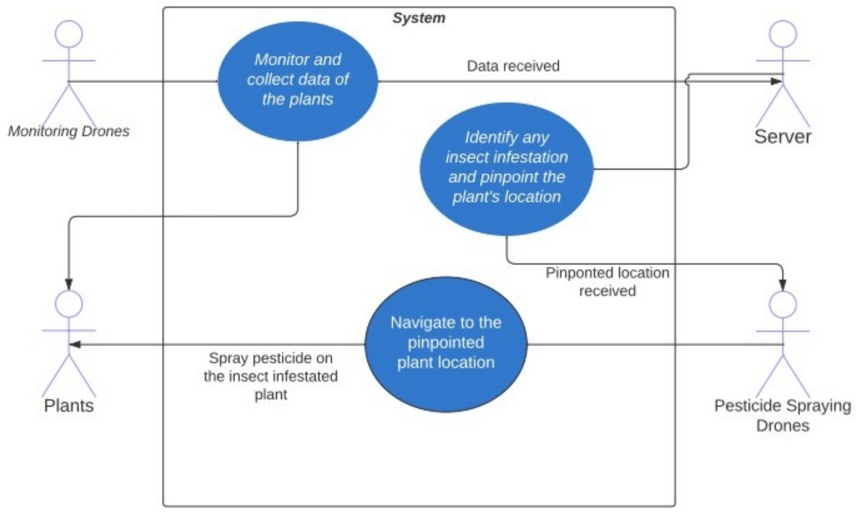


Fig. 2: Infestation Detection

2.3 Weed Detection (Charles Aresnal Okere)

In the case of weed detection the drone is actively involved in the monitoring and the collection of the data specifically on the field. This data is periodically being sent to the server to be processed. The server plays a vital role in ensuring that the data is properly processed and analyzed. The smart device or the monitor further pinpoints the specific weed locations.

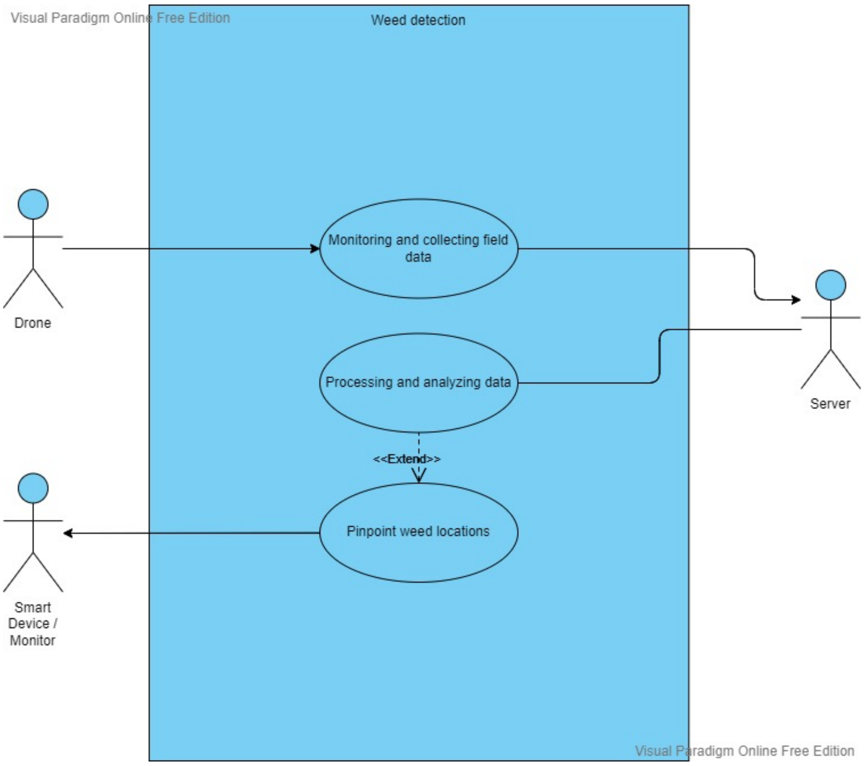


Fig. 3: Weed Detection

2.4 Disease Detection (Muhammad Subhan Khan)

When talking about disease detection as shown in figure 4, the farmer is communicating with the drone, the drone here records the information about the plants specifically related to disease, and after recording all of the information related to diseases about the plants, it summarizes the information, after the summarizing takes place, this specific information is then saved. The information is then collected for research purposes, and this research is carried out by an expert. The expert sends this information to a cloud which then processes the disease information and then classifies it into two categories one is a disease caused by the environment and one is caused by insects. In the following tapaal scenarios shown from figure 5 to figure 8, this information is displayed.

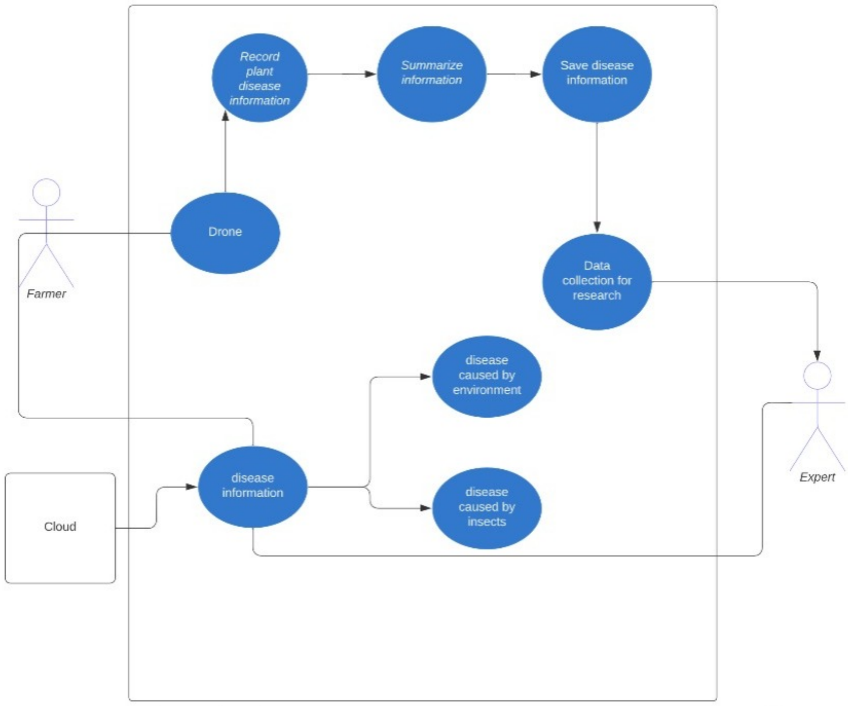


Fig. 4: Disease Detection

3 TAPAAL for the model-based specification

In this section, we modeled the scenarios using TAPAAL. Successfully none of the models had a deadlock in them.

3.1 Drone (Muhammad Subhan Khan)

This section talks about the simulation of the drone. In the simulation of the drone, the overall model of the drone is shown with all the properties satisfied. Moreover, after the drone gets the processed command from the operator then the drone will select which path to choose after the command is checked and it could be either infestation or weed detection as shown in figure 6. Here the drone chose the path of infestation detection as shown in figure 7, it detects infestation, and after the process has been done the infestation detection comes to an end as shown in figure 8.

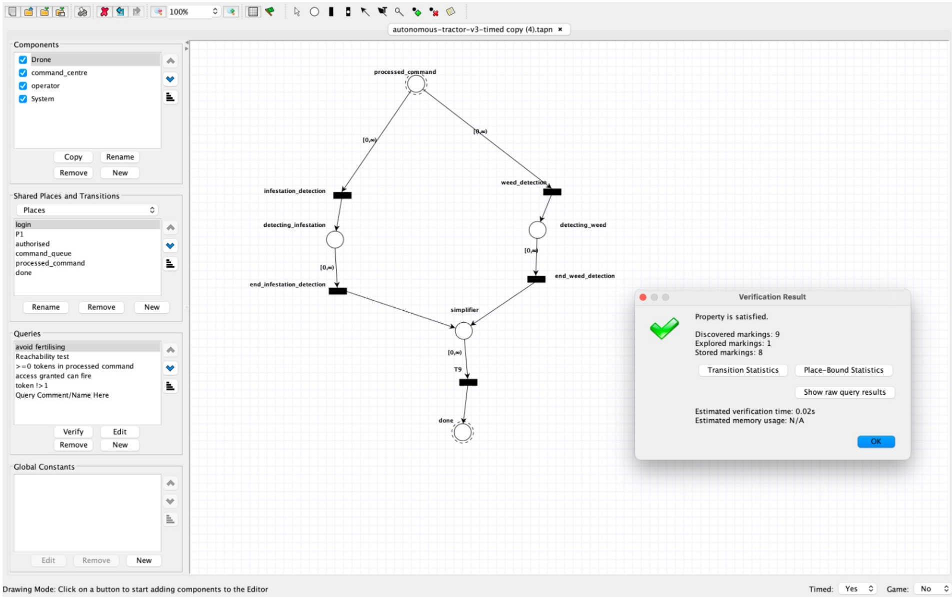


Fig. 5: Drone Model

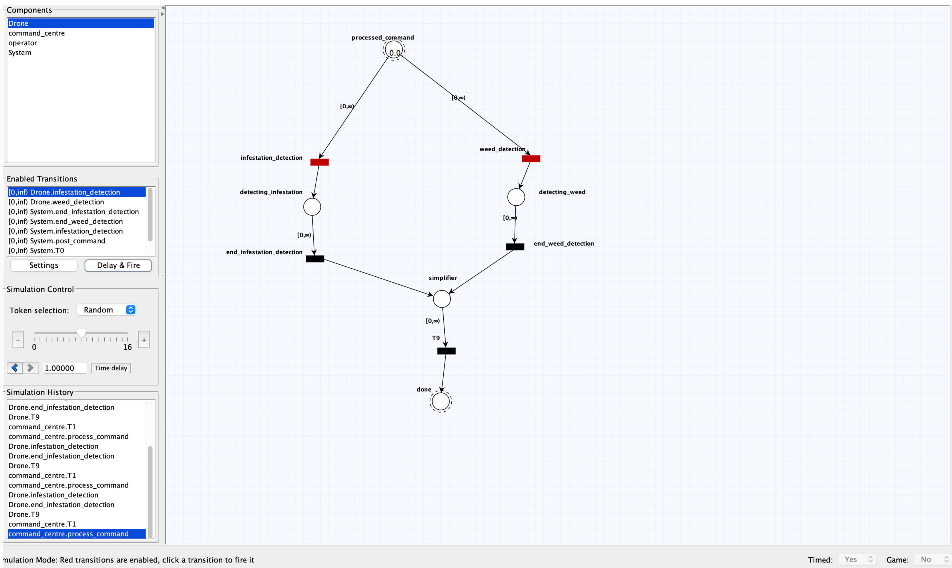


Fig. 6: Selection of path

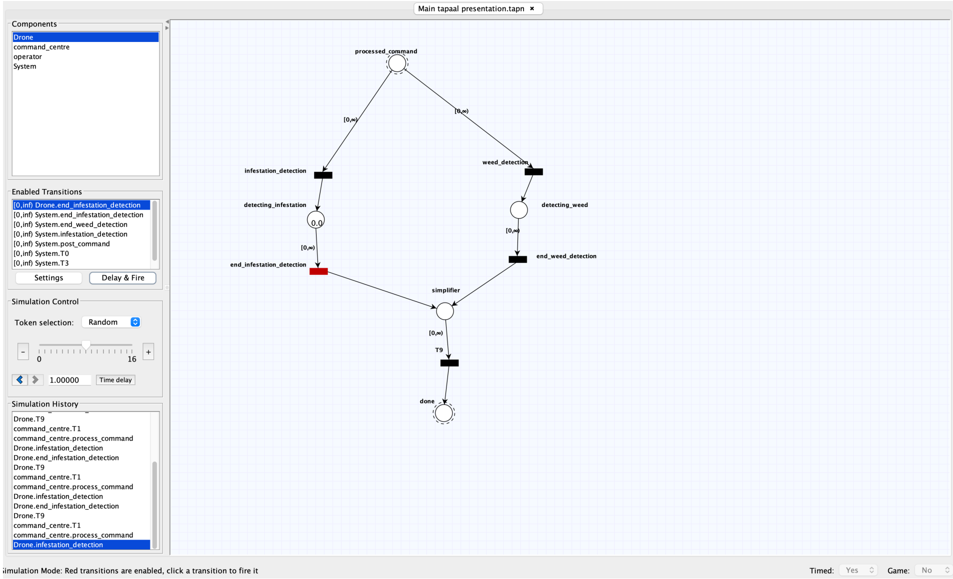


Fig. 7: Path selected

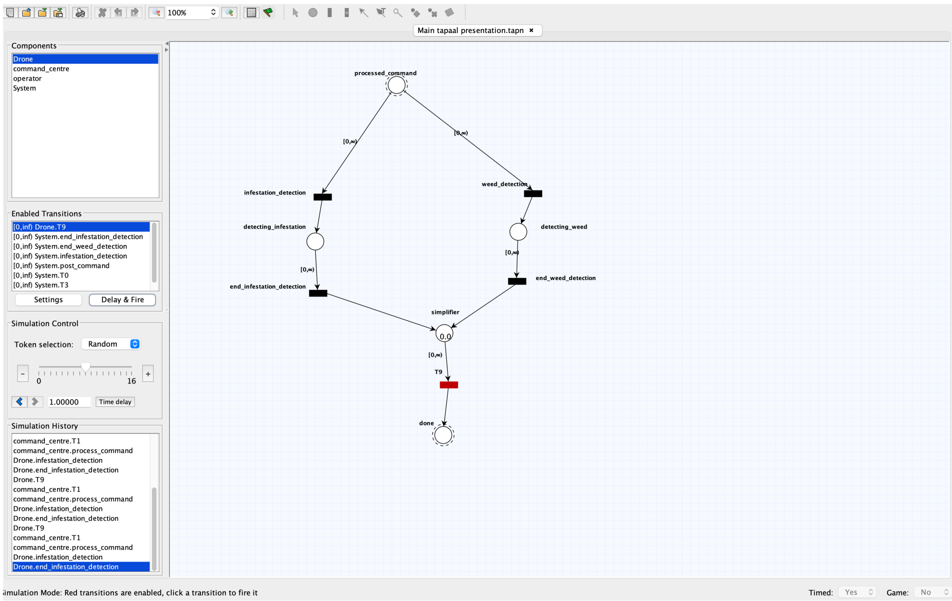


Fig. 8: Process completed

3.2 Command center (A B M Abir Mahboob)

This section talks about the simulation of the command center. After the user has successfully logged in, the user will be going through an authorization step as shown in figure 10, and then be authorized, afterward, wait for the command in the command queue for the command to be processed, and then finally the part that the command center plays will be completed as also shown and described in the activity diagram when verifying it the property of the command center was satisfied too as shown in figure 9.

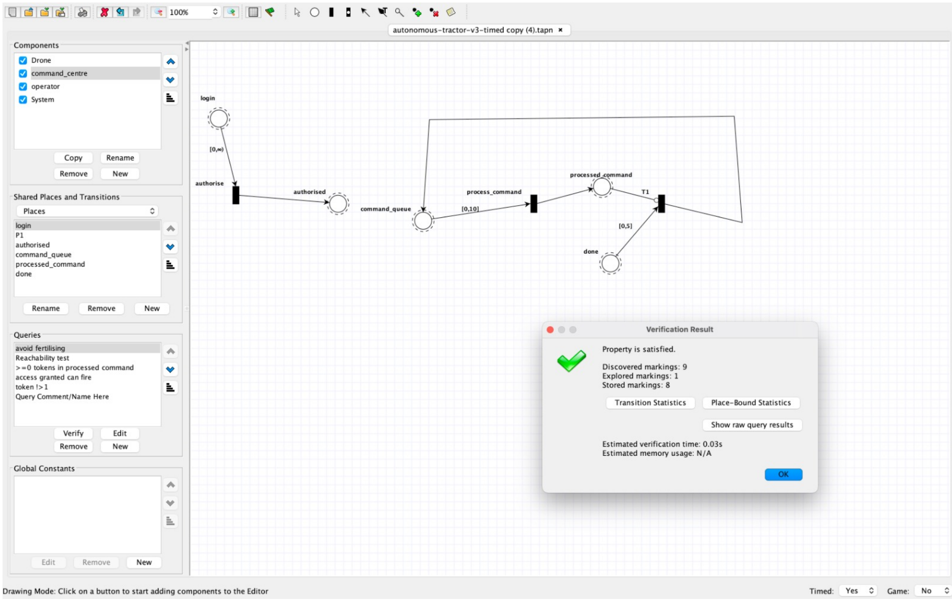


Fig. 9: Command Center Model

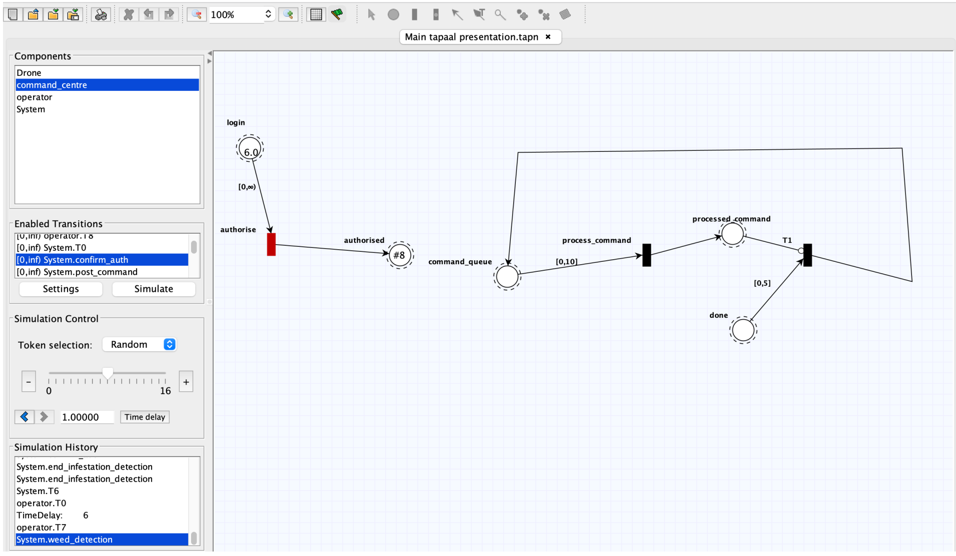


Fig. 10: Command Center authorization step

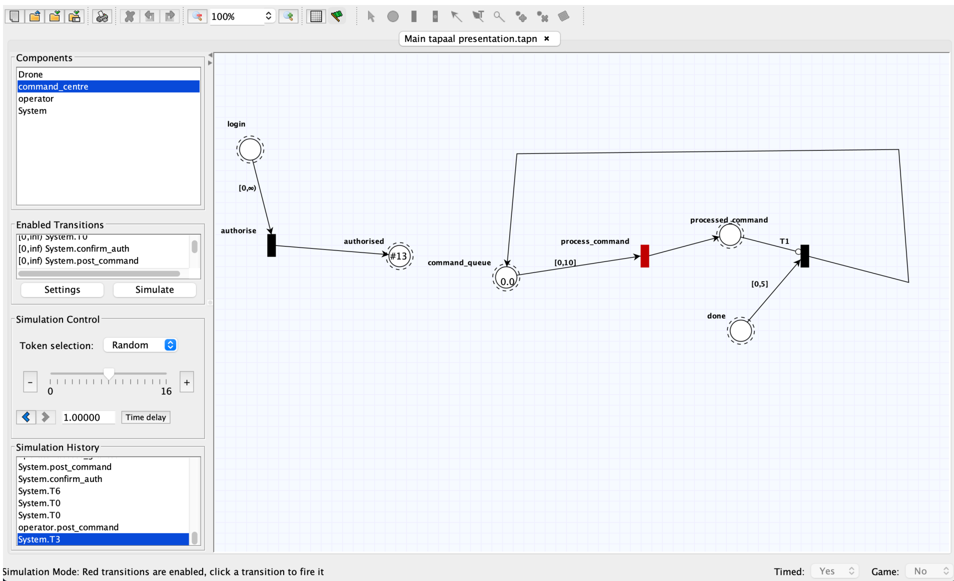


Fig. 11: Command Center process command step

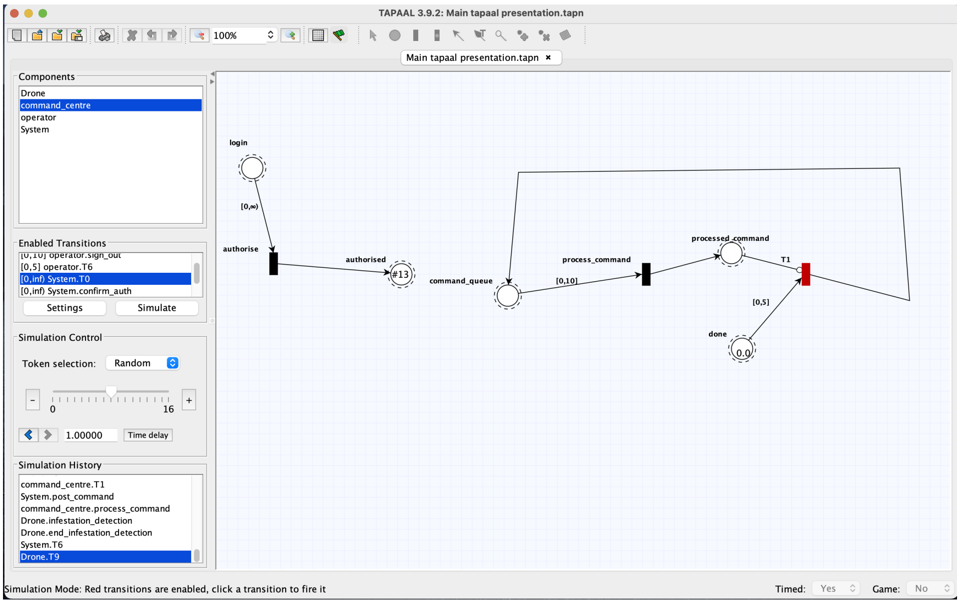


Fig. 12: Command Center process final step

3.3 Operator (A B M Abir Mahboob)

This section talks about the simulation of the operator, as explained in the activity diagram when verifying it this property was satisfied. This property is satisfied too in figure 13. The role of the operator was also involved in critical decision making of the system. After logging in, it waits in the queue for the access to be granted, after getting the access, it gets the input command from the command center and then the command is sent to the drone for the action to be processed, for example for infestation detection etc. After the process is completed by the drone, the data will be sent to the operator, and then the operator will finally sign out of the system.

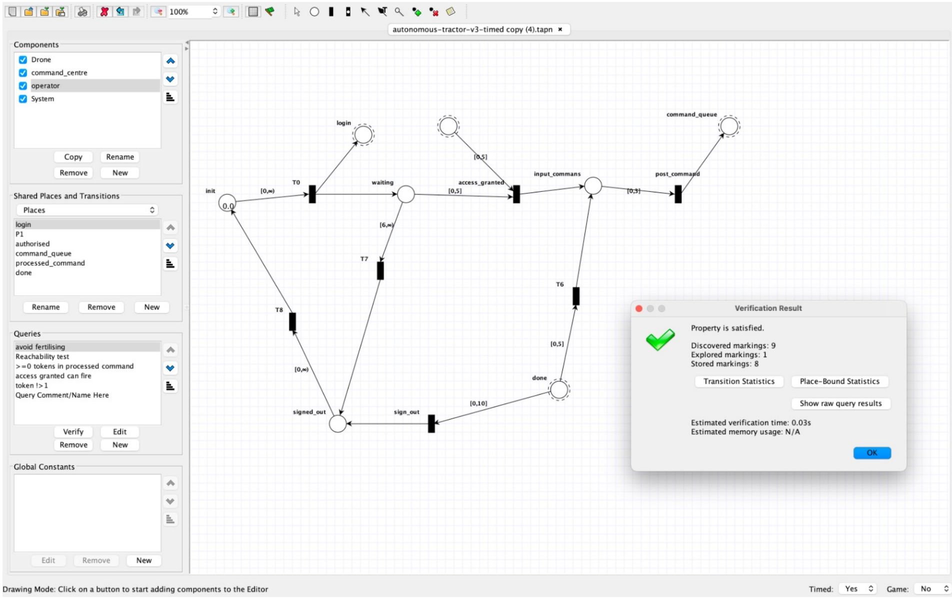


Fig. 13: Operator Model

3.4 System (Muhammad Subhan Khan)

This section talks about the simulation of the system. In the overall system, the overall property is verified here. This is where the integration of all the components took place, which included the part of the drone where it detected which type of detection it wanted to do whether it be infestation or weed detection. Moreover, also the role of the command center was to handle all the commands for example with the operator as discussed above in the operator section going in and out of the system as well as the queuing of the commands in a systematic way, for the smooth running of the system. The whole system was timed as well.

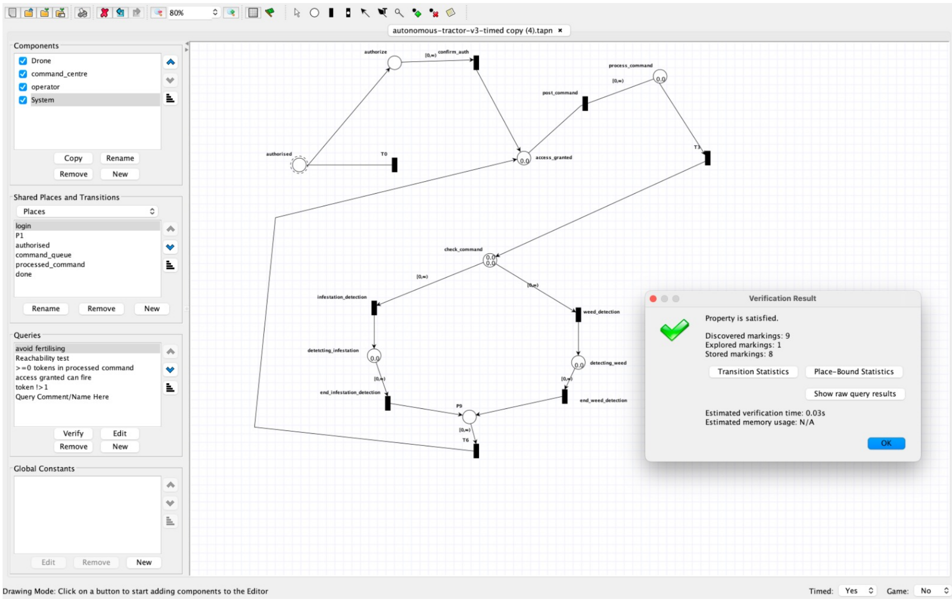


Fig. 14: System Simulation

4 Implementation of an appropriate algorithm (A B M Abir Mahboob, Muhammad Subhan Khan)

Prerequisites: Among the most powerful and well algorithms are convolutional neural networks (CNN), sklearn, numpy, and pandas Decision Tree. CNNs are frequently compared to how the brain processes vision in biological life forms. Convolutional neural networks are supervised Deep Learning algorithms that are most frequently utilised in computer vision and image recognition.

We implemented a Convolution Neural Network (CNN) on Plant recognition and weed detection.

Data-set Description:

Number of categories: 12, Number of images: 3484, Random Seed: 111

Used Python Packages:

NumPy : It is a numeric python module which provides fast maths functions for calculations. It is used to read data in numpy arrays and for manipulation purpose.

Pandas : Used to read and write different files. Data manipulation can be done easily with data frames.

TensorFlow: Deep neural networks are trained and inferred using this open-source software library.

Keras: For artificial neural networks, Keras grants a Python interface. For the TensorFlow library, Keras serves as an interface.

sklearn : In python, sklearn is a machine learning package which include a lot of ML algorithms. Here, we are using some of its modules like train test split, and accuracy score.

The following phases were performed in implementing CNN.

1. Building Phase: The dataset was processed using numpy arrays. Then we randomly shuffled the whole dataset.

2. Operation Phase: We splitted the data-set for testing and training purposes.

x train, x test, y train, y test = train test split(images, labels, test size = 0.2, random state = random seed)

The dataset is divided into training and testing on the line above. We are specifying the test size parameter's value as 0.2 since we are splitting the dataset in a ratio of 80:20, for training and testing. A pseudo-random number generator state called a random seed variable is utilized for random sampling.

Terms used in code :

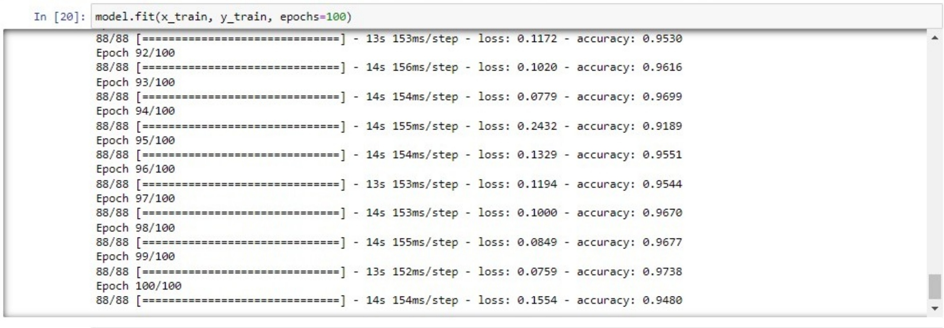
Accuracy score: The accuracy score is used to calculate the accuracy of the trained classifier.

Confusion Matrix: A confusion Matrix is used to understand the trained classifier behavior over the test dataset or validate the dataset

5 Implementation of a simulation environment (A B M Abir Mahboob)

The project's last phase involved identifying an appropriate environment and implementing the stated scenarios, including integration with the second phase. This task was executed in Jupyter Notebook utilizing the Python environment.

First, we imported all the necessary libraries (mentioned above). Using numpy arrays the data set was processed. Then the processed data was randomly shuffled and displayed. Now for training and testing purposes, we splitted the dataset into a train set and a test set with a ratio of 80:20. After the splitting process, we displayed the random images from the train set. We implemented the sequential model to train the training set. The attributes were set in such a way that any plant other than the sugarbeet plants are detected as a weed. After this process, we can view the total parameters, trainable parameters and non-trainable parameters. Then using the Adam optimizer and sparse categorical cross-entropy loss to compile and categorize them. We used 100 epochs to train the set. Each epoch got an average accuracy of 94 percent and took an average of 14 seconds to process (shown in Fig: 15).



```
In [20]: model.fit(x_train, y_train, epochs=100)
88/88 [==============================] - 13s 153ms/step - loss: 0.1172 - accuracy: 0.9530
Epoch 92/100
88/88 [==============================] - 14s 156ms/step - loss: 0.1020 - accuracy: 0.9616
Epoch 93/100
88/88 [==============================] - 14s 154ms/step - loss: 0.0779 - accuracy: 0.9699
Epoch 94/100
88/88 [==============================] - 14s 155ms/step - loss: 0.2432 - accuracy: 0.9189
Epoch 95/100
88/88 [==============================] - 14s 154ms/step - loss: 0.1329 - accuracy: 0.9551
Epoch 96/100
88/88 [==============================] - 13s 153ms/step - loss: 0.1194 - accuracy: 0.9544
Epoch 97/100
88/88 [==============================] - 14s 153ms/step - loss: 0.1000 - accuracy: 0.9670
Epoch 98/100
88/88 [==============================] - 14s 155ms/step - loss: 0.0849 - accuracy: 0.9677
Epoch 99/100
88/88 [==============================] - 13s 152ms/step - loss: 0.0759 - accuracy: 0.9738
Epoch 100/100
88/88 [==============================] - 14s 154ms/step - loss: 0.1554 - accuracy: 0.9480
```

Fig. 15: Accuracy

Then utilizing the test set we predict our model and display the result shown on Fig: 16 (actual and predicted plants).

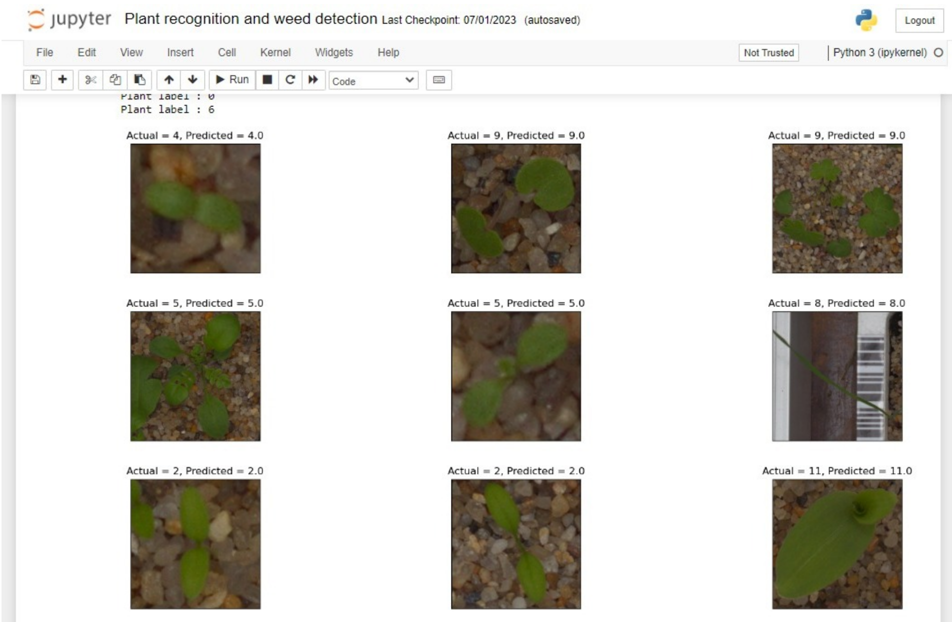


Fig. 16: Prediction: actual plants and predicted plants

Finally, we saved the model so we can add more datasets to train and predict. For a better understanding, our Jupyter notebook submission for this specific task and processes can be examined.

6 Conclusion (Muhamad Subhan Khan)

In conclusion, all the steps, have been explained in detail in this documentation. The project started with the modeling of the scenarios, which were then described in the use case and activity diagrams. The second phase included the modeling of the specific scenarios in Tapaal. The simulation environment was created using python, in order to properly specify the modeled scenarios and the average accuracy achieved was around 94 percent. It is a significant fact that the performance of information technology in precision farming is not static , rather its dynamic . Our future direction of work will be more in-depth in order to comprehend the core scenarios in precision farming and examine how farmers can carry out the varying levels of technologies and complexity in the future

7 Github History (A B M Abir Mahboob)

All the tasks that includes the use cases, diagrams, Taapal models and the algorithm were uploaded to github.

Github: `https://github.com/abmabir/Precision_Farming`

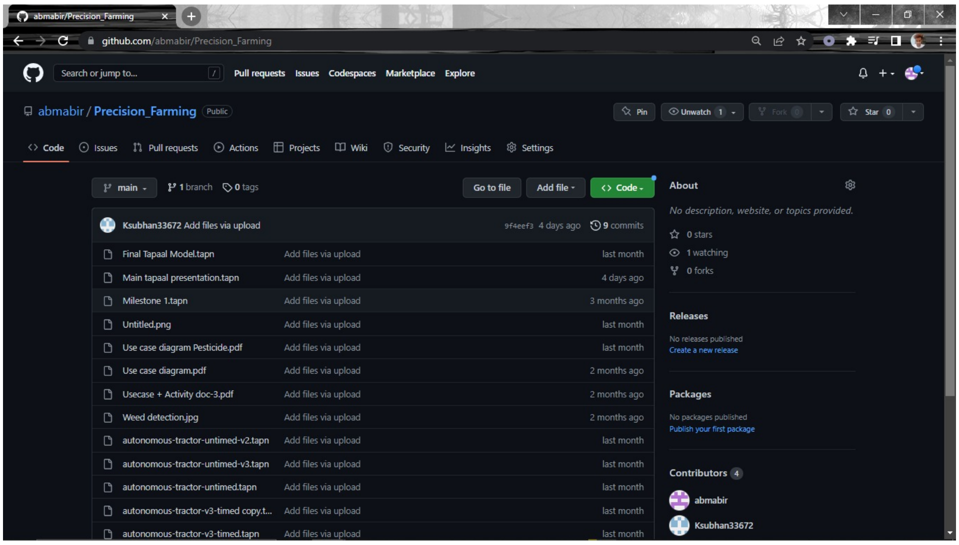


Fig. 17: Github History

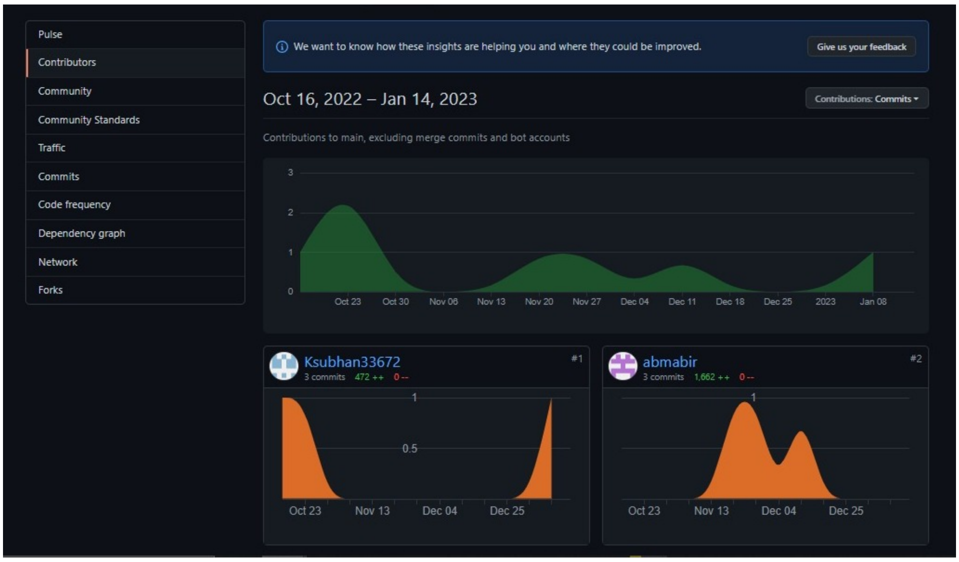


Fig. 18: Github History

8 Project Contribution/Team Management (A B M Abir Mahboob, Muhammad Subhan Khan)

8.1 Project Part

We tried to divide the workload of the project equally. Every member of the group made their own individual diagrams and use cases. Then those individual use cases were updated (if necessary), modeled and verified using Tapaal. For the identification of the appropriate deep learning algorithm we conducted several brainstorming sessions. After opting for the appropriate dl algorithm we integrated the overall solution including coordinated autonomous vehicles and dl approach. For this part the whole group worked together and contributed.

8.2 Documentation and Presentation Slides

The author's name is written next to each chapter of the documentation. The presentation slides were made by A B M Abir Mahboob and Muhammad Subhan Khan.

9 Declaration of Originality

We, A B M Abir Mahboob, Charles Aresnal Okere, Chiagoziem Cyriacus Ugoh, Muhammad Subhan Khan, herewith declare that we have composed the present paper and work by ourselves and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. we agree that our work may be checked by a plagiarism checker.

14.01.2023&Hamm - Team 6