Precision Farming

Tunde Oluwayemi Aluko[1], Enkeledi Mema[2], Fawad Murad[3], and Muhammad Moaz Amin[4]

Department of Electronic Engineering, Hamm-Lippstadt University of Applied Sciences

Contents

[1] tunde-oluwayemi.aluko@stud.hshl.de
[2] enkeledi.mema@stud.hshl.de
[3] fawad.murad@stud.hshl.de
[4] muhammad-moaz.amin@stud.hshl.de

Abstract: (Fawad Murad)

The idea of precision farming is to apply inputs into production (such as chemicals, seed and fertilizer) only when and where it is needed for the most cost-effective production. Basically, it provides a framework through which farmers can better understand and regulate what occurs on their fields. In this documentation, it is demonstrated that how different aspects of precision farming work and how we can get an idea of improvement in several places like how well we can train our system for better classification of different crops or detecting the problems using drones and the high-resolution map produced by internal system. Thus, the goal is how new technology now make it possible to put precision farming into practice in a real-world context in order to minimize costs and maximize benefits for increasing agricultural producing units' economic return.

1 Introduction And Motivation (Tunde Oluwayemi Aluko)

According to the United Nations[Na19], the world population will increase by 2 billion in 2050. To meet the ever-increasing population's food demand, farmers need to maximise their productions. In order to maximise production, precision farming is the future farming system that farmers need to adopt. Precision farming improves efficiency, productivity and cost.

Bahr et al.[Ba15] defines precision farming as a farming system that focuses on monitoring plants, animals and fields in real-time and provides responses. Precision farming often employs robotic systems like Unmanned Aerial Vehicles(UAV) to continuously observe and treat any part of the farm field or crops with its precise needs[Ra19]. These concepts improve farming productivity by optimising the inputs process, reducing labour cost, and increasing crop yield.

In this project, we have used the cyber-physical system (CPS) concept combined with deep learning algorithms to design and implement a precision farming prototype.

We first discuss our design approach by analyzing the system, after which we propose an architecture, then interpret our concept using use-case diagrams. We proceed to model our design concept in Petri-net with the aid of TAPAAL software as a support tool and use fastai[HG20], a deep learning framework built on PyTorch to model and train our deep learning algorithm. Our focus area of precision farming is weed detection.

2 Analysis of the System (Enkeledi Mema)

Precision farming (PF) is perceived as a system which goal is to increase the productivity by minimizing the resources. Usually there are different operation that occur in PF. After further requirements analyses, it is possible to split these operations in three main categories. The first is operation relates to the exploration. The second it associated with monitoring results. The last operation is to develop some strategies for field treatment. All these operation will be performed mainly by drones and partially from vehicles on terrain. Therefor, it is useful to build different use case based in specific scenarios. The first scenario is surveillance operation which is performed by a specific drone. These drones are called monitoring drones. The next scenario that needs to be taken in consideration is autonomously driving agricultural vehicles on terrain. During the work distribution, one drone might not be sufficient to fulfill an operation.

Therefore, several drones operations scenario is taken in consideration. To have a better control and information for the holistic system external system scenarios are beneficial[Ng17].

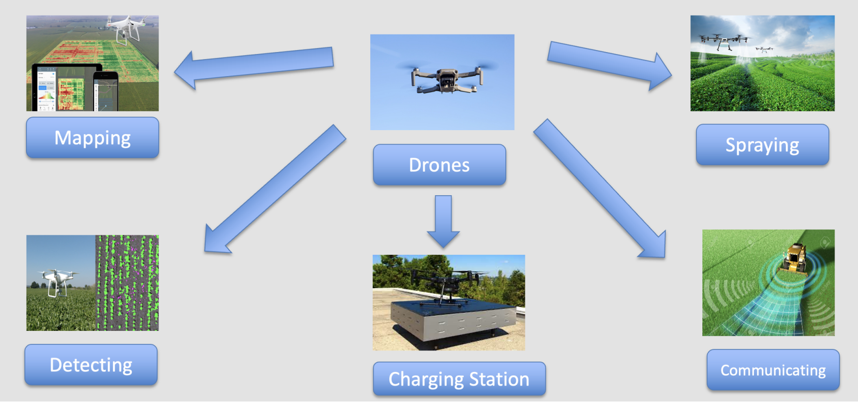


Fig. 1: Precision Farming System.

2.1 Use Case

2.1.1 Use Case for Monitor and Control of the Holistic System (Enkeledi Mema)

Based on the scenario for overall system monitoring the following use case diagram is generated.The purpose of this use case is to analyse and make visible all component of the system . Also external systems are included. In this use case is presented the communication that happens in between different actors.The first actor that has access to the system is the farmer. As it is shown in the use case the farmer can monitor the overall process.The farmer also can access the system and can perform manual adjustment. The other actor is an external system that can control the environment. This external system might be a person for example an agronomist. In the same time might be an institution or company that is interested in specific or partial information.This includes data processing or data computing.The next actor is a virtual system that provides the simulation in real time how the system is performing . Usually the virtual system compares the date in real time with the physical system. In case of a malfunctions it can generates alerts and provide information for the error that might occur.The next and last actor is the autonomous charging station.The charging station communicate directly to the drones.It can provides a dynamic charging for the drones based on specific priorities.

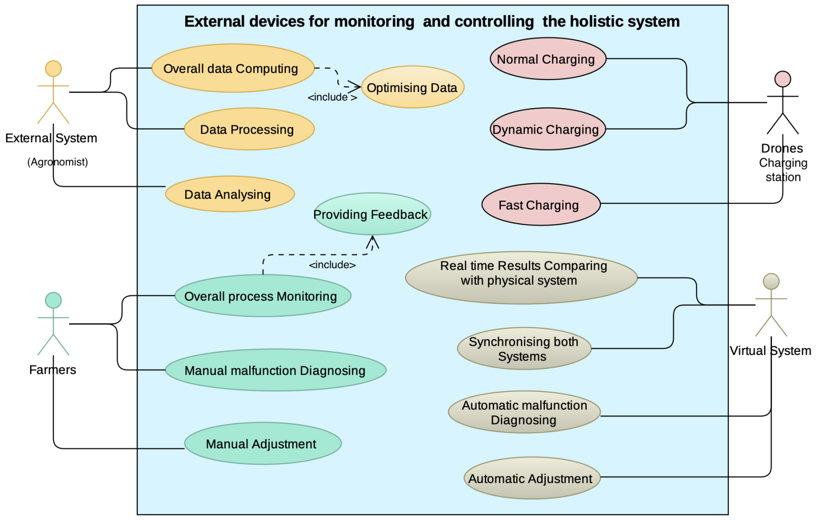


Fig. 2: Use Case of the Holistic System.

2.1.2 Use Case Specification of Monitoring by Drone (Fawad Murad)

In this scenario, we have the following actors and their use cases:

- Drone: (i) The first use case or goal of the drone in this scenario is to take several images by multispectral camera. (ii) The second use case of it is to record the imagery data.
- Internal System: (i) Internal system will overlap the pictures once the data is recorded by drone (ii) and then it will produce a high-resolution map of health of entire field (i.e 2400 times more detailed than satellite imagery).
- Farmer: (i) The first service of the farmer comes in the very beginning of this context and i.e. to set up drone path over the field to be mapped. (ii) to look into map using reflectance property (iii) Once the HD-Map is produced by internal system, then it will observe how well plants are reflecting the light from the Sun (iv) and another responsibility of the farmer comes at the end i.e. to use the analyzed information from Agronomist for the better growth. Like when and where to apply fertilizer or pesticide or how to manage irrigation.
- Agronomist: (i) The first use case of Agronomist is to scout and take samples of plant and soil (from the areas that are under some kind of stress) using drone data (ii) then analyse the samples. (iii) Once the samples are examined, it will inform the farmer about the analysis.

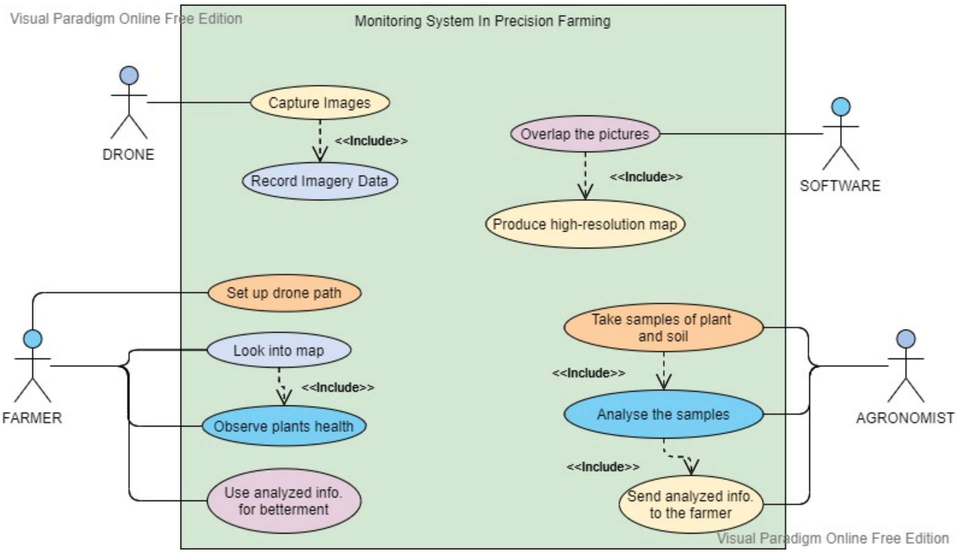

Visual Paradigm Online Free Edition
Monitoring System In Precision Farming
DRONE
Capture Images
<<Include>>
Record Imagery Data
Overlap the pictures
SOFTWARE
<<Include>>
Produce high-resolution map
Set up drone path
FARMER
Look into map
<<Include>>
Observe plants health
Use analyzed info. for betterment
Take samples of plant and soil
<<Include>>
Analyse the samples
<<Include>>
Send analyzed info. to the farmer
AGRONOMIST
Visual Paradigm Online Free Edition

Fig. 3: Monitoring System in Precision Farming

MAP DETAILS: These maps give us an early warning and allow us to solve the problems: (i) plants in green areas of map are extremely healthy (ii) plants in red areas of map are facing serious issues like drainage problems, pest infestation,etc. (iii) plants in yellow-orange areas of map are suffering unhealthiness from early stage which human eye cannot identify that easy.

2.1.3 Use Case: Multiple Drones (Tunde Oluwayemi Aluko)

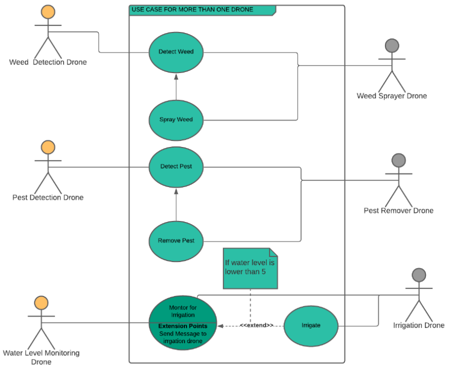


Fig. 4: Use-Case Diagram for Multiple Drones

A single drone has been used in many aspects of precision farming, such as spraying or monitoring extensive farmland; however if the farmland is a bit larger, it can be inefficient because it requires considerable time and energy. In contrast, when using multiple drone systems, it is possible to carry out cooperative works simultaneously (collaboration) or individual agricultural tasks on the assigned farmland (a division of labour).

Figure 4 describes the use-case diagram where we have six drones; weed detection drone works in collaboration with the weed sprayer drone. The same concept goes for the pest detection drone and the pest remover drone.

2.1.4 Use Case Specification of Autonomous Farming Vehicles (Moaz)

In this scenario the autonomous farming vehicles will carry out the farming operation of harvesting the crop and if there is some hindrance in the path obstacle avoidance manoeuvre will be carried out. In this scenario we have the following actors and their use cases. Figure 5 shows the Use Case diagram for this scenario.

- **Farming Vehicle:** Farming robot is the most important actor of this scenario and it is going to perform the on ground operation of farming and is also going to drive autonomously on the provided path data. The motor of farming vehicle which is responsible for the mobility of the vehicle is actuator without which the whole operation will fail as vehicle will not be able to move.
- **Farming Scientist:** The role of the scientist is to carry out manual operation of the farming vehicle in case there is a situation where autonomous driving is deactivated or is not accessible then farming scientist is going to operate the vehicle remotely. Farming scientist is also the one who is going to provide the plant information to the vehicles so harvesting operation is carried out and all the pests are removed in time so the main crop is saved. One other role of the scientist is to provide route data which along with the present drone data is fed to the robot so it can start moving. The scientist defines the boundaries which the robot cannot cross and will automatically turn off if breached.
- **Drones-Routing:** These are the drones which are collecting data that is required to map out the whole field. This data is updated in real time through wireless capabilities. These drones follow the trajectory of the vehicle and they also follow which driving path is taken by the vehicle. The drones also send back the data to the scientist so he can monitor any errors present in the autonomous operation.
- **Operational-Drones** (i) While the vehicle is moving around the field, Operational-Drones are acting as the vehicle's eyes. They are capturing high resolution images so that they can be compared with the algorithm present which will allow the harvesting operation to be carried out. (ii) These Drones also allow the farming vehicle to detect which plant is to be harvested and which is to be left aside as it has not yet fully bloomed. The secondary role of these images is to update the data-set currently

present so the complete process of reaping the crop has minimal errors. (iii) As multiple vehicles and other unforeseen obstacles can be present in field at any given time, Operational-Drones detect that and send the signal to the farming robot so that it can execute the obstacle avoidance manoeuvre.

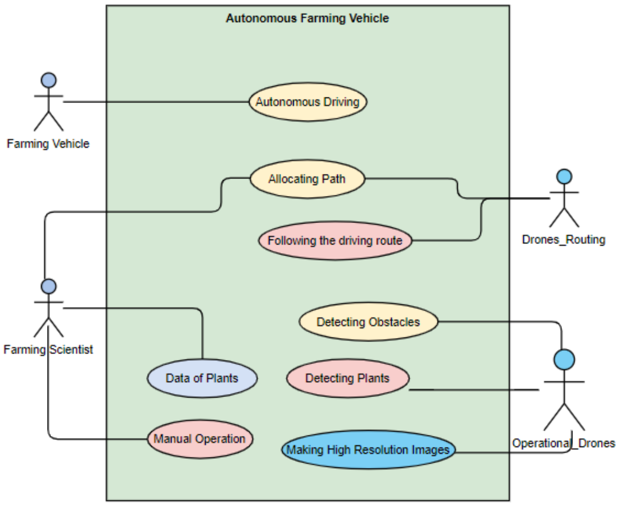


Fig. 5: Autonomously driving Farming vehicles

2.2 System Architecture (Enkeledi Mema)

Based on requirements and specifications that were achieved through use cases and scenarios,the overall system architecture was developed.

- Sensor layer is the first basic layer. In this layer are included all the types of sensor that are required to develop the precision farming.The sensors are in a wide verity depending on the purpose of usage.But it is possible to slit in terrain sensors that monitor the soil parameters and other conditions. The second group is the sensors used in different autonomous vehicles.
- The next layer is the network layer.This layer is responsible for all communication behaviours. The communication can be separated in long range data transitions and communication between devices.For data transmission it is possible to use WiFi, 5G etc. For communication between devices it is possible to use Bluetooth, Zigbee etc. This layer is considered as a bridge for upcoming layers
- The Upcoming layer is the service layer. This layer is responsible for all data set manipulation.With other words in this layer are performed data management.In it is

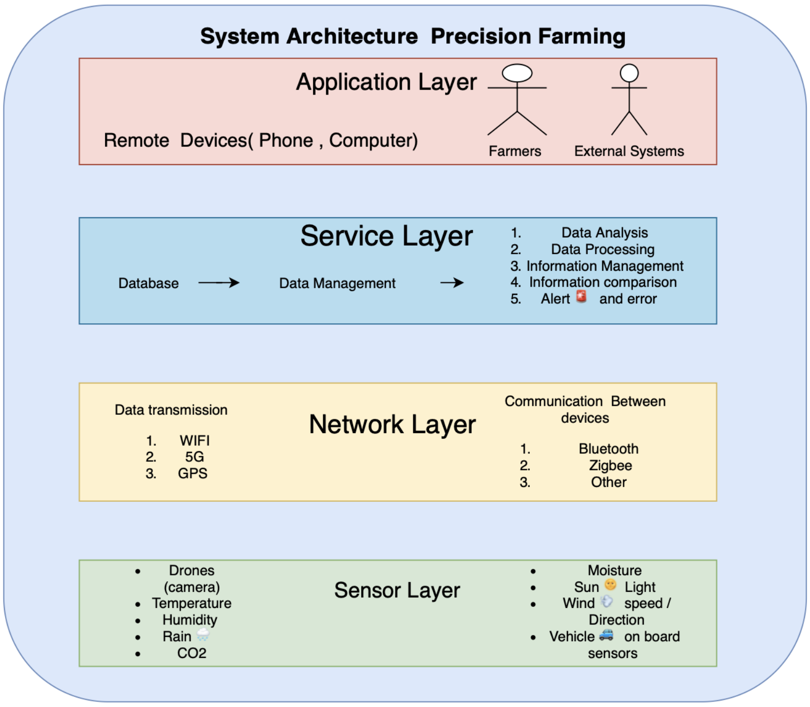


Fig. 6: System Architecture.

included data processing and analysing as well. Furthermore errors and alerts are part of this layer.It assist to switch to the next above layer.

- The final and top layer is application layer.This layer is responsible for granting access.Remote devices are included as well. A farmer or external system(Agronomist) can communicate with the system through this layer. Usually the application layer is responsible for input and output information for the system.

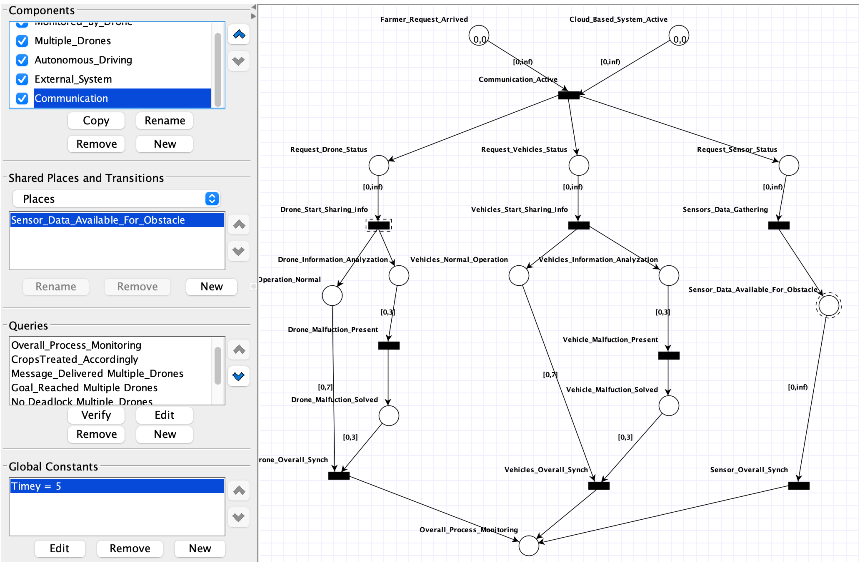
3 Modelling and Verification (Fawad Murad)

We used TAPAAL for modelling and verifying our Timed-Arc Petri Nets in order to have a better illustration of how precision farming thoroughly works. A graphical editor for developing TAPN models, a simulator for experimenting with the built nets, and a verification environment that automatically answers logical inquiries expressed in a subset of CTL logic (EF, EG, AF, AG) are all included in the TAPAAL tool. TAPAAL has the ability to check/verify model-related queries. There are a lot of options in the query dialog.

To begin, you can give your query a name at the very top of the window. TAPAAL will convert your model to a Network of Timed Automata (NTA) and then utilize the UPPAAL verification engine on the resulting NTA to accomplish the actual verification.

3.1 Overall Communication Implemented in TAPAAL(Enkeledi Mema)

Based on the use case for the holistic system,the next step is to model the system.To activate the communication the cloud based system should be activated and some requests from a farmer or an external system have arrived.Depending on the type of request, we can receive different information regarding the drone status,vehicles on terrain status and sensors on the field.In case of the drones status there is a shared transition with multiple drones TAPAAL model as it is shown in the picture. After the Information is received the farmer can check the status and for any type of malfunction. In case of such errors he can make manual adjustments based on conditions. In the same way the farmers can check the status of the autonomous vehicles on the field. In case of malfunction the can analyse and solve the malfunctions manually. The same scenario applies to sensors in the field.In this case there is a shared place with autonomous driving TAPAAL model.The farmer has access to the sensors and in case of malfunction it is possible to make adjustments or calibration manually.



Components

☑ Monitored_by_Drone
☑ Multiple_Drones
☑ Autonomous_Driving
☑ External_System
☑ Communication

Shared Places and Transitions

Places

Sensor_Data_Available_For_Obstacle

Queries

Overall_Process_Monitoring
CropsTreated_Accordingly
Message_Delivered Multiple_Drones
Goal_Reached Multiple Drones
No Deadlock Multiple Drones

Global Constants

Timey = 5

Fig. 7: Overall Communication Modelling and Verification.

3.1.1 External System for Charging Station(Enkeledi Mema)

As an external system,the smart charging station is modelled. First of all the smart charging stations should be running and waiting for drones requests. Drones can communicate directly with the charging station in case they need to be recharged. In case of an incoming drone request the smart station check if the station is available.If the station is available ,for the drone is assigned a time slot. This process is dynamic. This means that the queue is decided based on the priorities and can be updated.The next step is to allocate a time slot for charging. After the charging is done ,the drone is able to fly over the field again. This model shares the same charging done transition with multiple drone TAPAAL model.After the charging process is finished the queue is updated. Regarding the new incoming request new priorities are placed.The process can be repeated and the charging station is capable to operate autonomously.

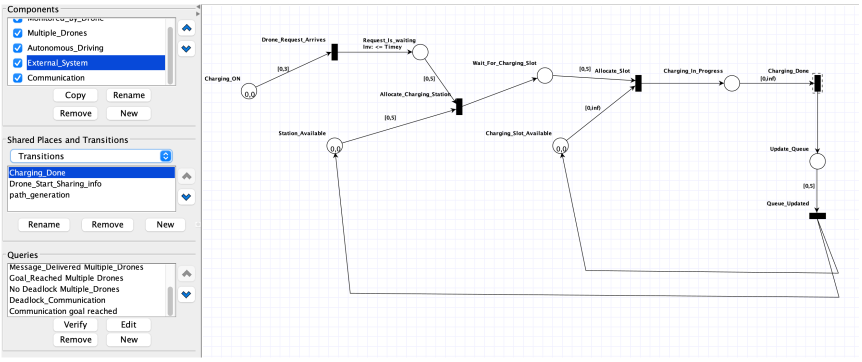


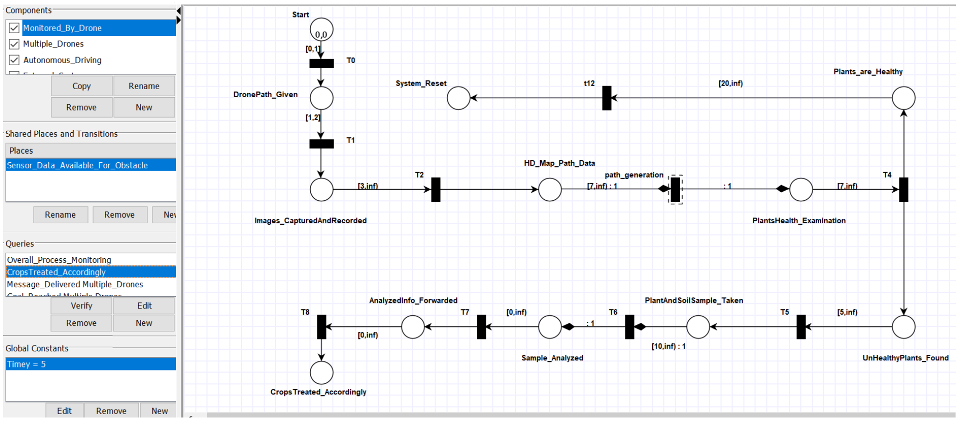
Fig. 8: External System (Charging Station) Modelling and Verification .

3.1.2 TAPAAL Models Verification (Enkeledi Mema)

The next step after modelling is verifying. Verification are very important because they ensure the system functionality,reliability and safety.In both TAPAAL models is verified if there are any deadlock. Due to analysing phase is proofed that the models do not contain any deadlock that might compromise the system behaviour.In the external system model it is important to check if a specific transition or place is reached.This type of verification is done for the charging done state in the external system.By using the same approach it is checked if the communication is reached. Another verification regarding the overall system monitoring is performed as it is shown in the pictures.

3.2 Tapaal Model of Monitoring by Drone (Fawad Murad)

In the beginning of this scenario, the drone path is given by the farmer. This path contains the information of the area of field to be monitored. Once the path is introduced, the next place in this model is for capturing images, in which the drone is capturing the photos of that specified area and recording the data of those images. The next place is for the generation of high-resolution map/path which is done by the internal system or software. After this place, there comes a transition where the information of that path is shared with one of the places in another interlinked scenario "Autonomous Driving" because that information is used in further places of that scenario.



Components

☑ Monitored_By_Drone
☑ Multiple_Drones
☑ Autonomous_Driving

Shared Places and Transitions

Places
Sensor_Data_Available_For_Obstacle

Queries

Overall_Process_Monitoring
CropsTreated_Accordingly
Message_Delivered Multiple_Drones

Global Constants

Timey = 5

Fig. 9: Modelling and Verification of Monitoring by Drone

Now, the farmer will look into that map details generated by the internal system in order to observe plants' health. We can see that there are two cases after this place in our model; one case will be considered when no unhealthy plant is found then our system will reset and the process can restart whenever it is desired, the second case will be considered when the farmer finds unhealthy plants in that specific area of the field. As the unhealthy plants are found, the farmer and agronomist will scout and collect plant and soil sample from those areas where there are unhealthy plants. The agronomist will then analyze the problem through those samples in lab. Once, the samples are analyzed, the agronomist will share the analyzed information with the farmer and the farmer will then treat the crops accordingly.

3.3 TAPAAL Model of Autonomous farming Vehicles (Moaz)

In the TAPAAL model of this scenario, various places and transitions are involved, some of which are shared and some are standalone that execute on their own. As the scenario begins the motor of the farming vehicle is turned on, simultaneously the vehicle tries to establish a connection with the Drones present in the field and the farming Scientist so that the autonomous ability of the farming vehicle can be enabled. Once the autonomous mobility feature is enabled the vehicle waits for the **HD-Map-Data** produced by the Routing-drones which are capturing real-time images which in turn is producing the path on which the farming vehicle is going to move. This particular transition in the model is shared with another scenario ”Monitored by drone” in the system.

As the vehicle begins to move on the allocated map it now moves to carry out the harvesting operation. In the pathway, if it encounters any obstacle at this stage it starts executing its obstacle-avoidance-maneuver which in this scenario is to wait for pre-defined time units present in the sensor data that is available on all possible obstacles in the field. As this is a shared place this information also comes from Operational-drones which are updated in real-time. It then moves to the next place **”Obstacle-avoided”** and then converges back to the normal transition of new path allocation. After this transition, the normal Farming operation is carried out and an output report is sent to the initial Communication-place for the report building of the current progress of the field. This whole system is modeled in Figure 10.

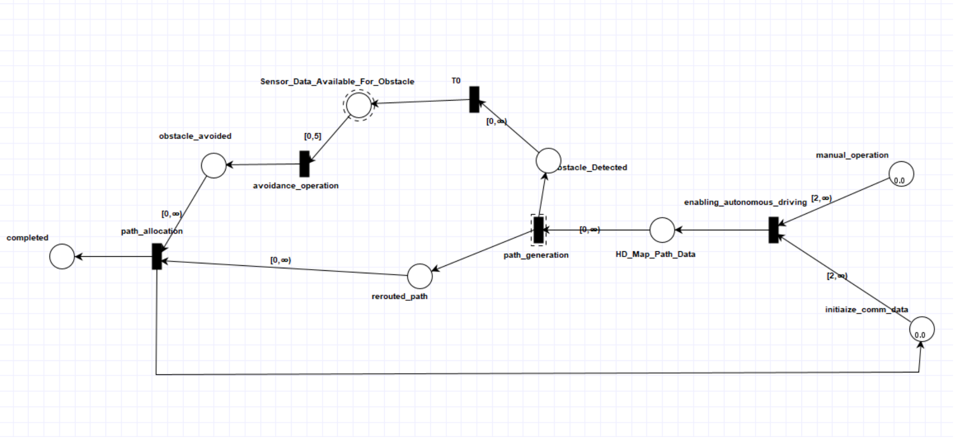


Fig. 10: Modelling and Verification of Autonomous Farming Vehicles

3.4 Tapaal: Multiple Drones (Tunde Oluwayemi Aluko)

We model our use-case diagram in TAPAAl as seen in figure 4. The weed detector drone first goes to the charging station; after the charging is done, flight mode is activated and starts flying over the farmland to detect weed. Once the weed is detected, the weed detector drone sends the weed path to the weed sprayer drone within three to five-time units. The rest of the model is self-explanatory. We then tried to verify some part of the model, too, as seen in figure 12, and the overall verification was satisfied.

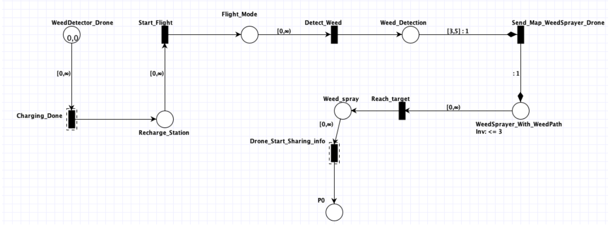
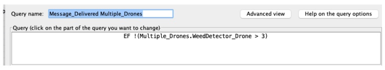


Fig. 11: Modelling of multiple drones TAPAAL



Query name: Message_Delivered Multiple_Drones

Query (click on the part of the query you want to change)

```
EF !(Multiple_Drones.WeedDetector_Drone > 3)
```

Fig. 12: Verification of Weed Detector in TAPAAL

4 Deep Learning Approach for Precision Farming (Tunde Oluwayemi Aluko)

The Deep Learning algorithm excels in recognizing patterns in enormous amounts of data. Object recognition from images is accomplished utilizing three or more layers of artificial neural networks, each extracting many features.

In this project, we have used fastai as our framework and Jyputer Notebook as the environment for training our training. The GPU server configuration is as follow:

GPU: NVIDIA-SMI

Version: 460.106.00

Driver Version: 460.106.00

CUDA Version: 11.2

This section will describe the dataset we used, the algorithm, analysis of our training and the result.

4.1 Dataset (Fawad Murad)

In deep learning approach, the first and most essential thing was to build a dataset on which the training of our system was based. Images of 12 known types of plant; ’black grass’, ’charlock’, ’cleavers’, ’common chickweed’, ’common wheat’, ’fat hen’, ’loose silky bent’, ’maize’, ’scentless mayweed’, ’shepherds purse’, ’small flowered cranesbill’ and ’sugar beet’ were supposed to be present in dataset. Even though, a collective data of these types of plant is already available in the given Jupyter notebook but we collected our own data from different sources by using different tools, like Python library, Kaggle, directly from Google images, etc.

First, we downloaded images individually as the collection of 900 images was distributed among four members. There were also few inefficient images among the whole collection of data we downloaded in bulk, so we had to double-check every image in order to observe its quality. We named each image file from 000 to 899 after gathering the data. Then, we created a CSV file in which semicolon (;) was used as a separator. The headers id and label appear in the first row (id;label), the second row has the number 000 and the label (000;black grass) and the 001 and the accompanying label (001;fat hen) are in the third row, and so on. The flight paths of all drones were also logged in a CSV file and one file was assigned for one drone. For example, each file contains the squares that the drone travels through (000;001;002;003;004). In the end, the dataset was uploaded as single zip file named with group name.

4.2 Algorithm (Tunde Oluwayemi Aluko)

Figure 13 shows our approach to weed detection using a deep learning algorithm. First, our drone detects the object and uses the CNN algorithm to classify the plant as a weed or some other plant. We would further describe this concept a bit further.

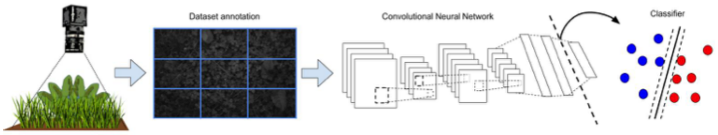


Fig. 13: Deep Learning based approach to weed recognition[KTN19]

Convolution Neural Network (CNN) has layers such as Convolution, ReLU Layer, Pooling, Fully Connected, and others. This model works in the same way as neurons in the human brain, with each neuron taking an input, doing some action, and then passing the results to the subsequent neurons. The images would be compared piece by piece using CNN. CNN compares images piece by piece using a feature-based algorithm. CNN employs the weight matrix to extract specific features from the input image without losing information about the image's spatial organization.

Transfer learning reduces the number of parameters in a model by applying a piece of the trained model on a familiar task in the new model. Transfer learning is a machine learning technique that uses a previously trained neural network. So, when we use the new model to analyze our original dataset, we effectively leverage the previously extracted features and train the model with our dataset. Furthermore, because the feature extraction phase does not need to be trained, we may train the model with minimal resources, datasets, and time[HTP19].

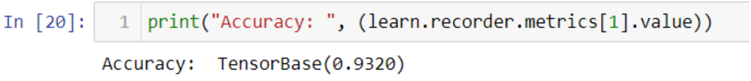
We have used RestNet[He16], a transfer learning model based on ImageNet[De09] this project.

4.3 Analysis (Enkeledi Mema)

After the deep learning algorithm implementation and model training,further analysis were done. In these analysis is included the wrong images classification. By identifying which images are classified wrong in a higher amount,it is possibles to perform further optimisation on the learner and training phase. If it is necessary, further pre-processing optimisation can be done on the data set. It was checked carefully the confusion matrix and other graphical information to find a suitable error rate and to decrease it. Our goal is to train our model in the best way. By analysing the confusion matrix carefully, it was noticeable a huge wrong classification of lose silky bent and black grass. Both plants in a young state were very similar and the model had difficulties in classifications. Therefore, we did optimization on the data set as well we performed more epochs and other data manipulation offered by fast-ai library and pytorch. Another function was used to make interpretation top losses. By analysing in details the function output, it was possible to check the prediction that was made ,what actually the image represents, the loss rate and what is the probability for a wrong classification. This was a really important tool to show visually the images that were classified wrong. Also to check the abilities that our model offers in image recognition and classification, we decided to take a random image from web. We tried to classify this image by using or model. In such kind of experiment we got relatively good result and the probability to receive a right classification was more than 90 percent.

4.4 Result/Prediction Moaz

After the overall implementation of the whole data set final result that we achieved is shown in Figure 14. By using different architectures, we achieved different results. The best result was achieved by using **”restnet152”**. As the model became more familiar with the data set the overall accuracy rate in each epoch increased. After 15 successful epochs, we managed to get an accuracy of 94% which is quite optimal considering we used a medium-sized data set that was compiled in a not-so-easy manner.



```
In [20]:  print("Accuracy: ", (learn.recorder.metrics[1].value))

          Accuracy:  TensorBase(0.9320)
```

Fig. 14: Accuracy of our Model

5 Conclusion Moaz

At the beginning of this project, we started with an abstract and basic idea about farming. After moving to the specific scenario allocation whole project started to make sense as we all could come up with our Use-Cases. Then we developed specific use cases that suited each scenario. After our use-cases were solidified we modeled our use cases and then connected them simultaneously where each scenario was co-dependent on each other. At this stage, we used different verification techniques to check for various deadlocks and tried to come up with a solution that will be a final working solution for all our linked scenarios. In the end, by implementing deep learning in classifying different images and using different algorithms in differentiating between elements of various data sets we became more familiar with Python and its libraries as a whole. As this project involved using a modeling tool TAPAAL we also acquainted ourselves with various verification techniques. This project gave us a real-time idea about how actual ”Smart Farming” in the industry is being executed. The overall experience of this whole project was really good for us all as it opened another door of knowledge about Precision Farming. We learned new methods in which farming which is the backbone of every food industry can revolutionize.

Declaration of Originality

We, Tunde Oluwayemi Aluko, Enkeledi Mema, Fawad Murad, and Muhammad Moaz Amin herewith declare that we have composed the present paper and work by our self and without 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 has not been submitted to any examination body and has 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.

Bibliography

[Ba15] Bahr, Claudia; Forristal, Dermot; Fountas, Spyros; Gil, Emilio; Grenier, Gilbert; Hoerfarter, Rita; Jonsson, Anders; Jung, Andras; Kempenaar, Corne; Lokhorst, Kees et al.: EIP-AGRI Focus Group: Precision farming. 2015.

[De09] Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai; Fei-Fei, Li: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee, pp. 248–255, 2009.

[He16] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778, 2016.

[HG20] Howard, Jeremy; Gugger, Sylvain: Fastai: a layered API for deep learning. Information, 11(2):108, 2020.

[HTP19] Hasan, Mosin; Tanawala, Bhavesh; Patel, Krina J: Deep learning precision farming: tomato leaf disease detection by transfer learning. In: Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE). 2019.

[KTN19] Kounalakis, Tsampikos; Triantafyllidis, Georgios A; Nalpantidis, Lazaros: Deep learning-based visual recognition of rumex for robotic precision farming. Computers and Electronics in Agriculture, 165:104973, 2019.

[Na19] Nations, United: , Department of Economic and Social Affairs, Population Division. World Population Prospects 2019, 2019.

[Ng17] Nguyen, Vinh; Vu, Quyen; Solenaya, Oksana; Ronzhin, Andrey: Analysis of main tasks of precision farming solved with the use of robotic means. MATEC Web of Conferences, 113:02009, 01 2017.

[Ra19] Razaak, Manzoor; Kerdegari, Hamideh; Davies, Eleanor; Abozariba, Raouf; Broadbent, Matthew; Mason, Katy; Argyriou, Vasileios; Remagnino, Paolo: An integrated precision farming application based on 5G, UAV and deep learning technologies. In: International Conference on Computer Analysis of Images and Patterns. Springer, pp. 109–119, 2019.