Precision Farming

Team-3[1]

Contents

[1] team-3

Abstract:

Exponential rise in the growth of population and a dire need of qualitative as well as quantitative food production give boost to consider advanced agricultural techniques that will result in more bounteous surplus. To effectively employ the advanced agriculture techniques, this paper proposes a unique set of instructions that wholly based on deep learning approach together with electronic engineering. Algorithm is designed for Image Recognition of various plants. Moreover, Petri nets and use case diagram were used to model the entire system In addition, other necessary equipments are discussed such as autonomous vehicles, sensors and drones. Hence, giving a motivation to implement the idea in real world scenario.

1 Introduction

Agricultural outputs mainly depend on the interaction between the its elementary features that includes, seeds, water and fertilizers. Monitoring all these inputs is very vital in order to get good results. Green revolution has been very successful over the years in terms of the crop production; however, usually due to poor managements, a lot of natural resources go to waste and environment is badly effected as a consequence. It is very important that the mankind move towards sustainable productivity, in order to limit the damage to the environment. Moreover, even the developing countries are facing a shortage of skilled labor, hence directly effecting the amount of agricultural production. The integration of modern technology has become absolutely a necessity, in order to overcome the issues mentioned before. Technological advancement has enabled and empowered the farmers to manage their agricultural activities in a more sustainable manner. Another emerging problem is that population is rising to much that requires a greater productivity of food. To meet the requirement of the food based on the number of people, it is essential to consider effective ways to produce food on the same area of agricultural land but with more quantity while maintaining the quality. Precision Farming is often termed as the information based technological farm management system, that helps to identify, analyze, and coordinate the variability within a farm, aiming towards higher profitability, and sustainability. The main goal of precision farming is to manage and distribute the resources keep the cost and sustainability into consideration. [Si10]

As shown in Figure 1, the benefits of precision farming maybe segregated into three main categories, that includes, Economical, Management and Management. GPS(Global Positioning System) is at most of the times considered as the main element of this technology. GPS basically allows to locate the exact location in real time, and that is done with the help of satellites orbiting the earth. GPS basically enables the system to map the soil and crop measurement at any given time. Other important devices used for the purpose of precision farming includes, Drones, remote sensors and variable-rate fertilizers. [Si10]

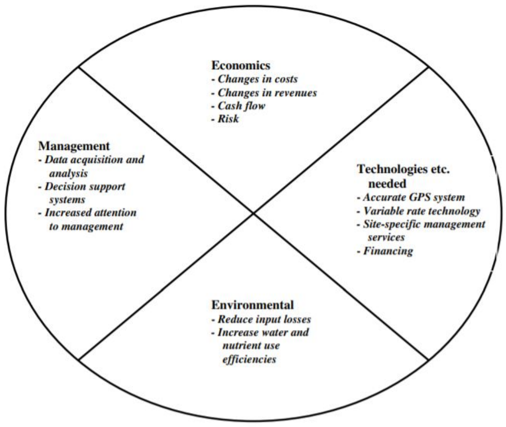


Fig. 1: Features of Precision Farming

[Si10]

Implementation in our project involves various aspects of Precision Farming Techniques such as, Drones, External Monitoring and Control, Autonomous Agricultural Vehicle and Agricultural Areas that cannot be controlled by Drone alone. To illustrate these scenarios we have implemented several Engineering Approaches that includes Use-Case Diagrams, Petri Nets and Graphs. Eventually, the project depicts a way to train and learn the project with the aid of a dataset in csv file which is based on 900 images. These 900 images belong to 12 differnet plants. Based on FastAI Library, Deep Learning Algorithms are programmed to give a vivid analysis and results that how accurately our model has trained. Hence, giving a direction to apply the concepts in real world problems. We believe that our approach would definitely gives fruitful consequences.

2 System Engineering

The following Use-Case diagrams were produced by Team-3 as result of Task-2

2.1 UAV Drone Monitoring -Mustafa Touqir

Precision agriculture, which utilizes GPS and big data to manage crops, has been touted by drone proponents for years as a method to boost crop productivity while addressing water and food shortages. Unfortunately, until recently, drones did not have a substantial impact on agricultural methods. Drone applications in agriculture and precision farming have received a lot of attention recently. Drones delivering agricultural intelligence for both farmers and agricultural consultants are changing agricultural practices, from the ability to image, recreate, and analyze individual leaves on a corn plant from a height of 120 meters to getting information on the water-holding capacity of soils to variable-rate water applications[Ve15].

Each drone has six subsystems. Each subsystem has a number of tasks that perform specialized activities related to a specific functionality. An UML Use Cases Diagram showing the UAV functionalities is presented in Figure 2.

Based on an examination of the Drone's functionalities: Movement Control is in charge of monitoring and controlling the engines and steering systems, such as flap actuators. It is made up of two jobs. The first is known as the Movement-Controller, and it is responsible for doing the calculus that must be performed in the actuators and engines. The Movement Encoder job is in charge of sampling and encoding the actual values in the engines and actuators, which will be utilized as feedback data for the Movement Controller task.

Navigation controls the UAV's movement directions and transmits control information to the Movement Control sub-system. It is made up of the tasks Route Control and Target Get Persuit. The first does the computation to direct the UAV via predefined way points, while the second does the same but for dynamic way points that vary in response to a moving object.

The communication subsystem is divided into two parts: LongRangeCom and ShortRange-Com. The first enables connectivity with pair communication nodes over vast distances (in the scale of kilometers), while the second provides connectivity over small distances (in the order of meters). Both rely on a third component, known as a Codec, to encode and decode transmission data using cryptographic techniques.

Collision Avoidance is made up of two tasks: collision detector and collision avoider. The first identifies potential collisions with other UAVs in the fleet or with non-cooperative flying objects, while the second calculates how to avoid the collision and sends the results to the Movement Control subsystem. .

The MissionManager and the Coordinator are the two duties assigned to this subsystem. The first controls mission-related information such as necessary data and mission policy, while the second coordinates with the other UAVs to minimize surveillance area overlap.

Image Processing collects analog visual information and digitalizes it. It carries out 5 major functions. The first is Camera Control, which is in charge of camera movement, zoom and focus control of IRC and VLC, and antenna orientation of the SAR. The second component is the Coder, which converts analog data into digital data. The Compressor, which compresses digital pictures, is the third component. The second component is the Reflectificator, which is in charge of the rectification as well as the reflection in the X and Y axes of the radar picture.The fourth function, Filter, is in charge of filtering the radar pictures to remove noise caused by the speckle effect. The final job, Pattern-Recongnition, is in charge of performing picture segmentation and pattern recognition from previously processed data.

2.2 External Monitoring and Control-Abdullah Zafar

An important part of the entire project includes the External Monitoring and Control of the entire system. As per figure A10 in appendix, this part of the system served as the gateway between the sensors/input devices and the actuators. As mentioned earlier, Drones are used for capturing field images, that are basically segregated in 12 different categories, and the target field was divided into 4 different regions, that was monitored using four drones.

The system works in a manner that the server receives an input from the four drones, and those pictures are analyzed using the image recognition algorithm. Moreover, input is also received from the sensors, that are embedded inside the field. The data that is received from these input devices is then processed, and the resulting decision or action is performed by devices such as harvestor. The farmer/client has the possibility of requesting the data as well, this basically includes data such as, optimum harvest time, plant growth status and plant segregation(image recognition). However, the tasks that followed along the project mainly involved image recognition, segregating 12 different plant types into different categories. The produced use case diagram(Appendix A10), further illustrates that the authorization might fail, incase a device outside the system tries to access it. This can help ensure the privacy and the control would be in the right hands. Moreover, the data that is received from drones and the embedded sensors is stored in server/database, that may be accessed at any point in time.

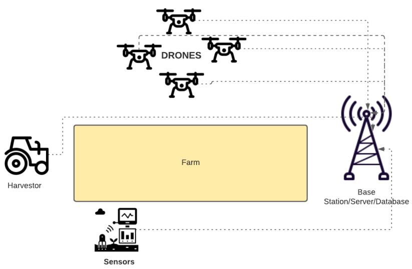


Fig. 2: Interactive system of precision farming

Figure 2, helps us to understand the relation and the route of communication between the devices. Moreover, this system maybe remotely accessed as well, it is not necessary one needs to be at the exact location in order to access data and make decisions. As mentioned in [Se97], hardware such as DGPS maybe used for the purpose of receiving a wide range of information, such as the latitude and longitude of a harvester or any other machine. As details help in determining the input parameters, that maybe later used to produce sustainable yield.

However, in this specific scenario, the main emphasis was laid on photo recognition and drones were used instead of DGPS. The total number of drones were four in this scenario that were placed over four regions of a field. Moreover, A linked was established between different module of the project in order to ensure synchronization of the system.

2.3 Autonomous Agricultural Vehicles - Dawar Zaman

One of the most important and trivial aspect of the project is the actual guidance of the Agricultural Vehicles. In this section we will discuss how agricultural vehicles communicate with the system and also with each other in order to perform their tasks optimally. As seen from the use case diagram in the appendix, A harvester (or any other agricultural vehicle) initially, sends request to a drone via the system the drone then starts gathering the data or if it has already recently gathered data, then, it provides the results in the form of a perimeter (a restricted area within which the agricultural vehicle has to stay throughout the operation) and a route (the path that the vehicle must take in order to properly cover the area). After receiving the perimeter and route details the harvester will communicate with other harvesters working in the same perimeter in order to further refine and optimize the route. After the route is received and refined the harvester will start working on the fields and it will keep working until the harvester has finished the working area allocated to it by the system or in case the harvester finds an obstacle in its path then it will simply change direction (go around the obstacle) and inform the system about its decision.

Various sensors and communication systems are involved in the operation of this system. For communication the system will consist of DSRC (Direct Short Range Communication) over IEEE 802.11p which will take care of all the V2V and V2I communication needs. The sensors that will be used of the agricultural vehicles will be intelligent cameras and intelligent IR cameras. These cameras will have the capability to detect an obstacle by processing it on powerful on board computers or sending the data to the server for remote processing. It is important to also detect obstacles from a ground level due to safety reasons and it is also important to use cameras instead of ultrasound sensors as ultrasound sensors might give a false result and might mix up a crop or plant with an obstacle. If 2 cameras are used on a vehicle in a sort of stereovision arrangement then it is also possible to detect the distance of the obstacle from the vehicle.

2.4 Several Agricultural Areas That Cannot Be Monitored By A Drone Alone - Syed Muhammad Saim

Precision farming can not be done only with drones and autonomous vehicles that are being deployed on a framing land. Sensors are necessary to gather data of the crops and lands from various parts of the land. In fact, sensors are essential building block of drones and autonomous vehicles. The data, which is gathered from various sensors, can be programmed

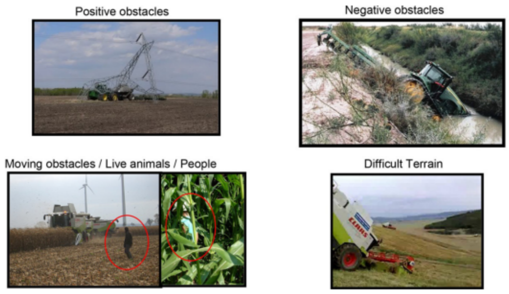


Fig. 3: Various Obstacles Detected from a ground level [Re16]

so that the system might learn and train based on deep learning algorithms. For this part of our research paper, I have focused on those agricultural areas that can not be controlled by drones alone that are required in precision farming. There are various sensors that can be discussed in detail, such as location sensors, optical sensors, electrochemical sensors, mechanical sensors, dielectric soil moisture sensor, but for the limitation of this paper, I have picked a pH sensor and explained it with the help of petri net models to give an extensive overview of external devices in precision farming. [SWW17]

Initially, my scenario, as shown in the Figure A13, is described by use case diagram. Drone, Autonomous Vehicle and System Administration are working as actors that are interacting with various use cases. The first use case is ’Get Data’ which is being shared with with Drone System Administration through association relation. This use case has also an include relation with ’Detect Variation’ use case that also has an include relation with ’pH Sensor’ use case. The concept here is that pH Sensor keeps on detecting the variations in the field and prepare the data that can be taken by Drone and System Administration. ’Get Data’ use case has also an exclude relation with ’Update Date’ which depicts that the data is updated whenever it is necessary. The next use case is ’Take Picture’ which has association relation only with the Drone actor. ’Take Picture’ use case has an include relation with ’Use Camera + LiDAR’. This idea here is to show that it is always necessary to use camera to take the pictures by the drone and LiDAR to guide the track for the drone. These pictures can then be validated as shown by the ’Validate Pictures’ use case that is only share with System Administration actor through association relation. ’Validate Picture’ use case also has an exclude relation with ’Plant Disease Detection’ use case. Only when a disease exist, it will be detected. One of the use case, i.e ’Command’, is being shared with all of the actors. The command is given by the System Administration to other actors to decide what to do next. The Autonomous Vehicle has another association relation with ’apply alteration’.

3 Petri nets

Timed Arced Petri Nets are a unique modeling and simulation tool that illustrates a scenario in self-explanatory way. Hence, allowing the models to verfiy as well. We have used petri nets to model our scenarios. Our models, connected together, were verfied as well using the Tapaal tool. These properties include EF, EG, AG, AF. These petri nets are as follows:

3.1 UAV Drone Monitoring - Mustafa Touqir

To describe the model UAV monitoring of fields we use timed petri nets. Figure 4 depicts the pictorial representation of rimed flow of the model. This model represents deployment of 4 drones from different onground-hubs. First deploymenmt is activated by the farmer or user. Then the drones are in action and they perform their designated tasks. The place that is referred as "deploymentïs basically a shared place and is interlinked with the next petri-net model in the system.

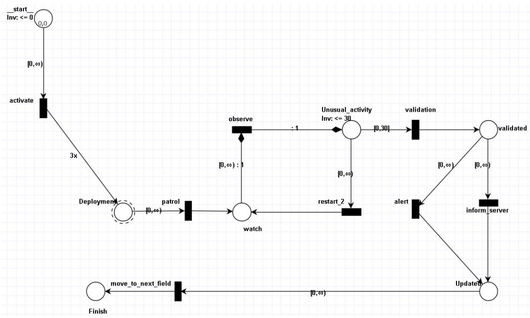


Fig. 4: UAV Petri nets

As mentioned earlier, Drones and the control and monitoring model are interlinked though shared place. The shared place is further linked into the harvester, hence forming a complete structure. Moving back to the figure, the token from deployment go through the transition in the watch place. here transport arch were used, in order to remember the age of the token. Here an iterative loop is followed, were the drones keeps a watch on any sort off unusual activity. In Case of an intrusion, validation takes place and the server and the system is alerted. This basically helps in maintaining a robust system, that also helps to secure the target fields.

Moreover, as per the given projects, drones are mainly utilized for the purpose of capturing photos. Moreover, There are four drones used in a field of region, that is also divided into four parts. The later sections explains those parts in a good detail. Petri nets is a good method used for model verification. It helps us to verify different properties, such as deadlock and EF,AF and AG etc. Share places helps to connect different petri nets, in order to formulate and verify an entire system in petri nets.

3.2 External Monitoring and Control- Abdullah Zafar

As briefly mentioned earlier, Petri-nets were made in order to give an overview of the entire system. Petri nets is basically a modeling language, that maybe used to model a distributed system.This system had four different modules, that were interlinked with one another, in order to make a complete system. The figure(5), illustrates on how the external monitoring and control system works.

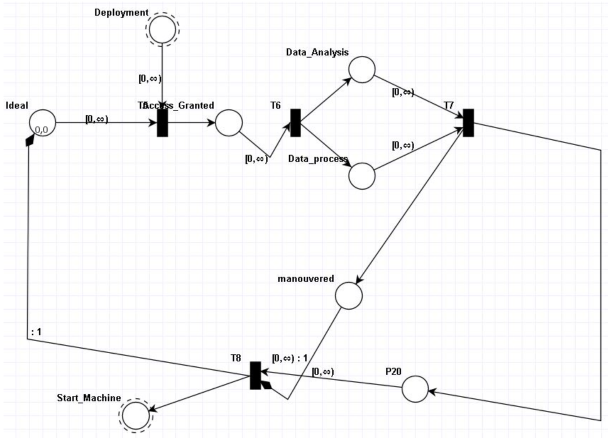


Fig. 5: External Monitoring and Control Petri nets

The petri nets were formulated in a manner, that it is interlinked with other parts of the system as well. Shared places were used in order to connect the different parts of the petri nets, and in order to formulate a complete system.The petri nets initializes from the place named as "Deployment", this input token is basically passed from figure(4.0) to figure(5.0). After initializing, the token pass through the transition ÄccessGranted", here the system insures that the access is being given to an authorized device. In a real-world scenario, this will help that the security and the safety of the system is not compromised.

moving on, the petri nets enter into data processing and Analyzing stage and this is where the learning algorithms and other computations are implemented. As shown in figure x, the analyzing and processing takes place simultaneously. Moreover, as per the given use case, transport arch was used in the end, before reaching the ideal state. Transport arch basically helps in producing the same aged tokens. The resulting output is then passed onto the a place referred as StartMachine. This place is again a shared place, same as the "Deployment", that was used initially. Start Machine basically sends an input command to the harvestor, hence initiating it. The continuation of the process is showed in (figure 6.0). Moreover, the model was also verified for different given properties in tapaal.

3.3 Autonomous Agricultural Vehicle -Dawar Zaman

The Petri−Net for the Autonomous Agricultural Vehicle in the Precision Farming Scenario works on the principle of shared places. The **Start_Machine** from the External Monitoring and Control initiates the process by giving the tokens to the **Request_Route** transition. This transition can also be seen in the use case diagram. After this transition the tokens are passed on over to **Gather_Data**. The **Gather_Data** tells the drone via the system to collect route information. After this the tokens are passed to **Provide_Route** transition. This transition is responsible for sending back the route information to the agricultural vehicle. After the **Provide_Route** transition the token has the possibility to go to two different places. It can go to the **Start_nth** place or it can go to the **Start_nth** or it can go to **Initialize_Operation**. In case there are more than one agricultural vehicles available for utilization the token will go to **Start_nth** from here it will go to the **Divide_Route** transition which is responsible for refining and further optimizing the route after this it will return to **Initialize_Operation**. After **Initialize_Operation** the token goes to **Follow_Route** transition. The **Follow_Route** transition shows that the vehicles have started working after this the vehicles will keep working until the perimeter is reached or an obstacle is found.

In case an obstacle is found after **Follow_Route** the token will move to **Obstacle_Found** then it will go to **Change_Direction** after this it will go to **Perimeter_Reached** and then to **End_Operation**. The **End_Operation** indicates that the agricultural vehicle has finished working on its allocated workspace and can either be allocated a new workspace or it can go back to where it was stored. After **End_Operation** the token will go back to **Start_nth** and the entire cycle will repeat it self.

The Petri−Net is verified with deadlock properties along with other Petri−Nets with shared places.

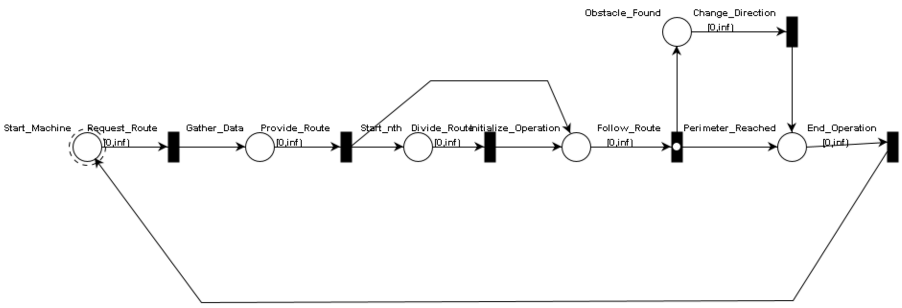


Fig. 6: Autonomous Agricultural Vehicle Petri Net

3.4 Several Agricultural Areas That Cannot Be Monitored By A Drone Alone - Syed Muhammad Saim

My Use Case Scenario is also implemented using Petri nets, as shown in Figure Tapaal - Syed Muhammad Saim, so that this scenario can be verified and validated. The Petri net model begins with pHSensor with age invariant of 2. This shows that data collected by the pH sensor is not more than the age of 2. This is done to continuously detect and get the data from the sensor within this age. For instance, if there is a variation in the agriculture land, which includes lesser minerals, lower moisture content, etc., then this could be detect and processed promptly.

The data gathered from the Sensor(s) can then be sent to System Administration/Server and Drone. This subsequent transition occurs after getting the data through the transition 'Get Data' and detecting it in the place 'Detect Data'. The transition occurred with the assistance of normal arcs that have infinite time interval starting from 0. 'Drone' is the place that has age invariant of 4. The data set from the sensor has to reach the 'Drone' place within the specified age invariant otherwise the system will not proceed. This check is done to keep the model robust. Moving further is the transition 'T11' that has an urgent sign. This is done to quickly get the token in the 'Deployment' place from preceding models that are connected with altogether. The place 'Deployment' that gets a token from the place 'Deployment' in the first Petri because all of the Petri nets are connected together and this place is being shared. This 'Deployment' place then connects to the 'take pictures' with the aid of transport arc, to maintain the age of the acquired token, after passing the transition T12 with an urgency. This urgency is shown by a white dot inside the transition T12. This occurs to quickly get the initial token from the first Petri net without any delay so that the model should synchronize with the whole system. After this, the model passes on to the place 'validation' through transition 'validate pictures'. This again occurs with the assistance of transport arc so that the integrity of the token remains constant. The 'validation' place then connects with 'system or server' place that has age invariant of 8 through transition T13. The model uses transport arcs from 'Deployment' place until 'system or server'. This is done to depict that this block of process should run in a very less time so that the pictures of the detected agriculture area could be captured without changing the state of that area. For example, if there is an insect or disease on the crops, it could be dealt swiftly without damaging the crop any further. Eventually, the data is saved and the model starts over again. The model can also be verified based on the various properties that include AF, EF, AG, EG, etc. There was no such deadlock during the simulation.

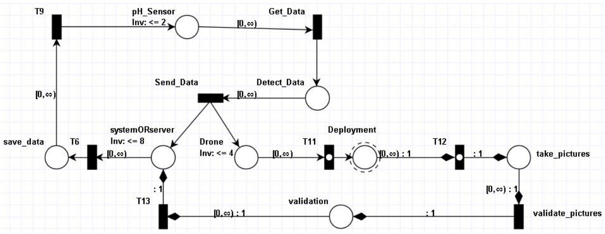
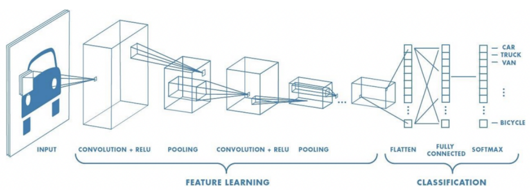


Fig. 7: Several Agriculture Areas That Cannot Be Monitored By Drone Alone

4 AI Model

Deep Neural Networks have been shown to outperform typical machine learning techniques for image categorization for several years. Neural networks give improved functionality for doing the same jobs more efficiently. As a result, most agricultural AI researchers are increasingly turning to deep learning models, which provide them with good performance on most fronts.

Deep Neural Networks have been shown to outperform typical machine learning techniques for image categorization for several years. Neural networks give improved functionality for doing the same jobs more efficiently. As a result, most agricultural AI researchers are increasingly turning to deep learning models, which provide them with good performance on most fronts[CKD21]. Example of a typical CNN architecture is depicted in figure below. This paper uses Resnet-152 architecture because of its low memory consumption in comparison to that of Mobinet which are more widely used for such purposes. Also, Resnet work better on smaller images which is usually the case in our plant seedling classification.



4.1 Tools and Frameworks

Our model to detect plant seedling is based on FAST AI deep learning library. It is a cutting-edge PyTorch-based system used for object identification, picture segmentation, and image classification[CKD21]. It performs quicker processing than its competitors and has built-in data cleansing functionality in the form of widgets. Another significant advantage is that it has a very basic workflow, which simplifies the debugging process. Fastai has a vision module that includes all of the operations required to construct a dataset and train computer vision models. One of the submodules named vision.data has a one-of-a-kind utility function called ImageDataLoaders, which accepts input from.csv files, picture directories, image lists, and so forth[HG20].

After then, the data may be divided into training, validation, and testing sets (if required). Another function provided by Vision.data is DataBunch, which organizes the training data into batches. These batches are sent consecutively to the training model. Other sub-modules, such as vision.transform and vision.learn, include functions for transforming/augmenting data and training data[HG20].

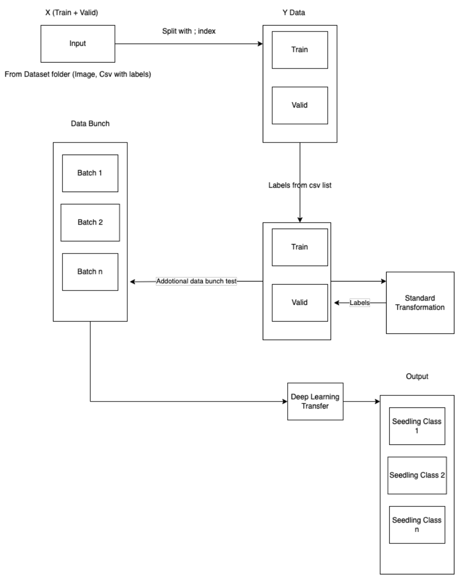


Fig. 8: Fast AI architecture for data pipeline[CKD21]

The DataBunch approach speeds up the transition from data blocks to training model architecture. After receiving data, 'vision.learner' offers all functions for training the model. As previously stated, the backbone architecture for transfer learning in this research is Resnet-121. Resnet121's dense layers are trained repeatedly on top of the convolutional base layers utilizing several epochs and suitable learning rates.

The 'lr find' approach is used to determine the appropriate learning rate. Fastai provides two techniques for tracking training and validation losses with the given learning rate: callbacks and earlyStopping. Over a given number of training repetitions, learning rates fluctuate evenly. The 'fit one cycle' strategy is to blame for this[HG20].

Stochastic Gradient Descent is used to train and fine-tune the layers (SGD). Fastai also allows you to freeze and unfreeze layers, which allows you to keep gradient descent under control. A lot of hyperparameter adjustment is done when training the model to get the optimum classification accuracy. Because these hyperparameters are part of the Fastai modules, they are version dependent[HG20]. For this model the hyperparameters used were following:

Random Resized Crop = 460
Batch Size = 32
Flip = False
Vertical Flip = False
Maximum Rotate = 460
Maximum Zoom = 32
Image Size = 24
Num-workers = 4

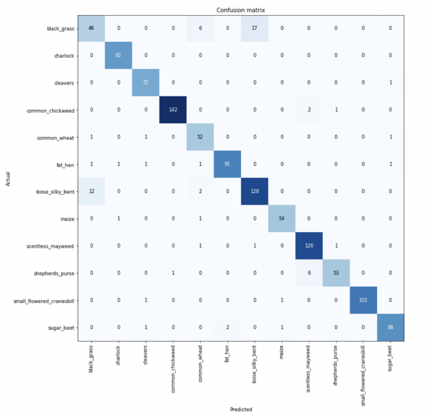
4.2 Results

After Training for 98 epochs, after which the learning rate decreases, with Resnet 152. The final model obtained a classification accuracy of 92.78 percent on the validation data set using the Resnet-121 backbone architecture, hyperparameter tweaking, and Fastai data cleaning procedures. Some of the performance metrics and their results on the test set are as follows:

Precision = 0.576

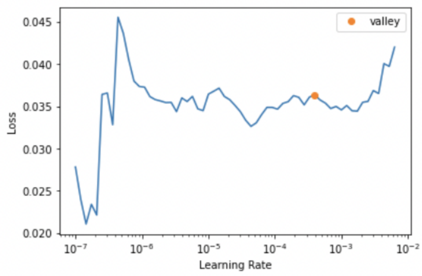
Recall = 0.531

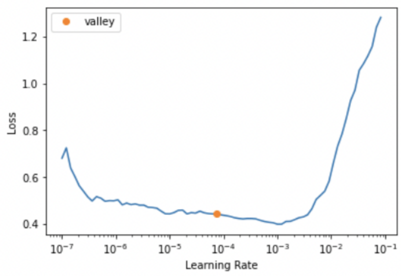
The ResnetLoss duringFinal Training for the validation set is given below:



The ResnetLoss duringFinal Training above shows that the majority of the mis-classified photographs belong to the same crop type. The biggest number of misclassified photos, for example, is 17 between Black grass and loose silky bent. They have a fairly similar appearance, therefore the mistake is acceptable. Another type of mis-classification is when the crop's condition is the same even if the crop itself is different. Black grass and common chickweed, for example, have been misclassified six times. They are both in the process of developing leaves.

Furthermore, given the distribution of classes in the original data set, there is little over-fitting in the final model since the training and validation losses begin at a high point and reduce to almost comparable values as the epochs proceed. The training and validation loss graphs are depicted as follows:





It is also clear from the second graph that data cleaning significantly reduced training and validation losses as compared to the first graph, which reflects losses prior to data cleaning.

4.3 Drones Integration

The initial assumption made before selecting the optimum number of drones was that the field is divided into 4 equal regions, As illustrated in figure xyz. Hence, it was decided by the team that the optimum number of drones to be used would be four. That meant that each region will be accommodated by a single drone that hovers over a set field of index. The required task was to recognize a batch of images, that are 900 in number and are divided into a 2-dimensional matrix. That meant that the total field of picture had 30 rows and 30 columns.

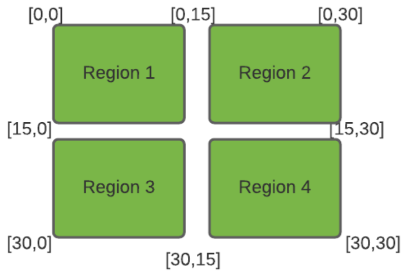


Fig. 9: Target field divided in four regions

The respective path of the drones were generated in the jupyter notebook, and those paths were exported as .txt files. That meant that a total of four .txt files were generated, whenever the code was simulated. Each of drones had a separate path, depending on the assigned region to the drone. The .txt file contained the 2D-array index of the path that it was flying over, the drones maybe assigned a different region, depending on the Use-case. The following code snippet was used in order to export the respective .txt file:

```
drone_one= open("drone1.txt","w+")
```

Compared to the real world scenario, this segregation of field of action will help in more efficiency. As mentioned earlier that a field of 900 plants of 12 different categories were covered, that means that each drone is responsible for capturing 225 photos. Apart from capturing photos and the usual work, drones maybe used for surveillance purposes as well. This will help the customer detect unusual activity over the target field.

4.4 Dataset compilation

The compilation of plant dataset was an important aspect of the project. The dataset consisted of 12 different species of plants:

1. Blackgrass
2. Charlock
3. Cleavers
4. Common Chickweed

5. Common wheat
6. Fat Hen
7. Loose Silky-bent
8. Maize
9. Scentless Mayweed
10. Shepherd's Purse
11. Small-flowered Cranesbill
12. Sugar beet

The dataset for these plants was very meticulously collected through various other datasets available on the internet and through scientific forums and databases. The dataset had many characteristics. It was dynamic however it did not contain multiple types of plant in one view. For example considering the dataset of blackgrass. The Blackgrass dataset contains many different pictures of the same plant at various ages and various angles however even though both blackgrass and wheat can grow in the same habitat and various datasets available online had mixed images of blackgrass and wheat. Care was taken that only those images are chosen where wheat cannot be seen. Similarly another proof of carefully forming the dataset can be seen from the Cleavers dataset where the images of the fruit of the cleavers plant were also taken into consideration in order to create a challenging dataset.

In all 75 images were gathered for each plant (900/12=75). A .csv file was also made in order to train the neural net on the dataset.

5 Conclusion

This implementation is a concrete overview of how Engineering Techniques can be applied to gain fruitful results.Our model is not 100 percent efficient and there is still need for further enhancements but the results from the Deep Learning Approach is an evidence that the model is ready for real world application.

6 Task allocation

1. Syed Muhammad Saim: Several Agricultural areaas that cannot be monitored by a drone alone(Petri nets, use case,AI Learning model)
2. Mustafa Touqir: UAV Drone Monitoring(Petri nets, use case,AI Learning model)
3. Abdullah Zafar: External Monitoring and Control(Petri nets, use case,AI Learning model)

4. Dawar Zaman: Autonomous Agricultural Vehicles(Petri nets, use case,AI Learning model)

It must be noted that abstract,introduction, AI Model and conclusion were written in collaboration and as a team. Practical work on AI model was also a combined effort and everyone was equally involved in every part of the AI project.

Bibliography

[CKD21] Chakraborty, Aditya; Kumer, Debarun; Deeba, K: Plant Leaf Disease Recognition Using Fastai Image Classification. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, pp. 1624–1630, 2021.

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

[Re16] Reina, Giulio; Milella, Annalisa; Rouveure, Raphaël; Nielsen, Michael; Worst, Rainer; Blas, Morten R: Ambient awareness for agricultural robotic vehicles. Biosystems Engineering, 146:114–132, 2016.

[Se97] Searcy, Stephen W: Precision Farming. A new approach to crop management. Texas Agricultural Extension Service, The Texas A&M University System, USA, 1997.

[Si10] Singh, Anil Kumar: Precision farming. Water Technology Centre, IARI, New Delhi, 2010.

[SWW17] Sureephong, Pradorn; Wiangnak, Patcharapong; Wicha, Santichai: The comparison of soil sensors for integrated creation of IOT-based Wetting front detector (WFD) with an efficient irrigation system to support precision farming. In: 2017 International Conference on Digital Arts, Media and Technology (ICDAMT). pp. 132–135, 2017.

[Ve15] Veroustraete, Frank: The rise of the drones in agriculture. EC agriculture, 2(2):325–327, 2015.

A Appendix

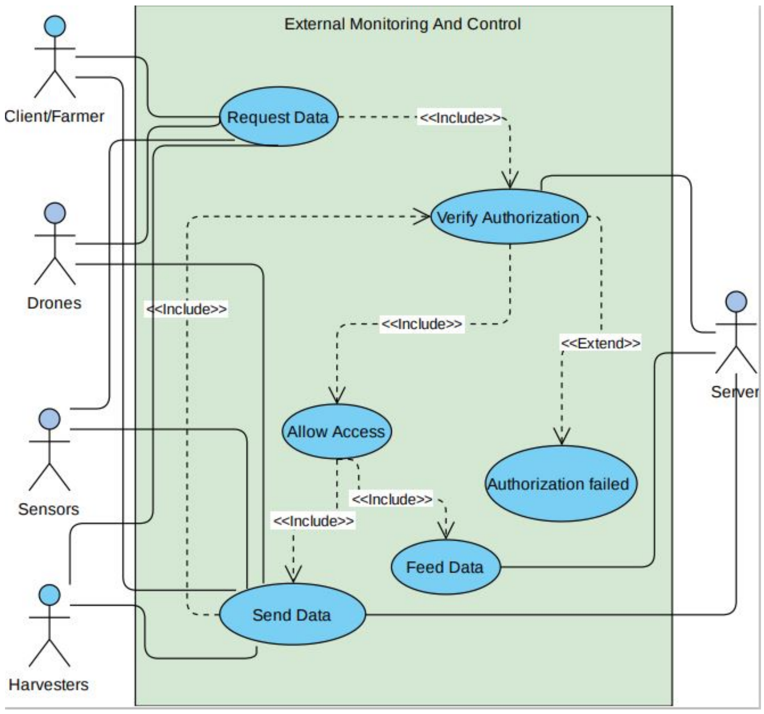


Fig. A.10: External Monitoring and Control Use-Case

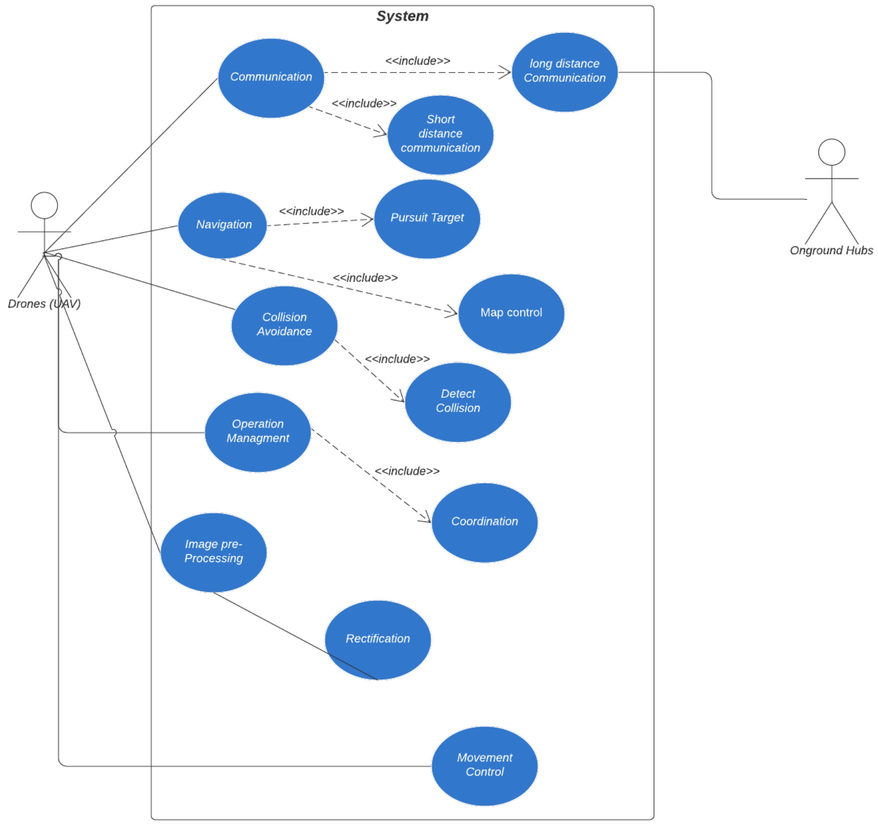


Fig. A.11: Use Case Diagram for UAV Monitoring of fields

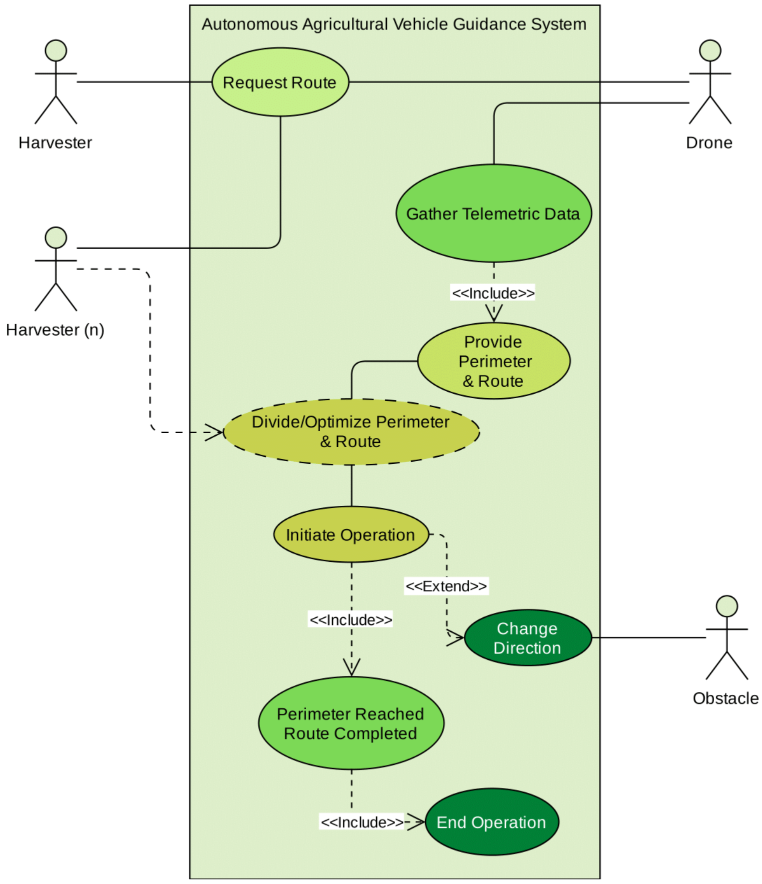


Fig. A.12: Use Case Diagram for Autonomous Agricultural Vehicle

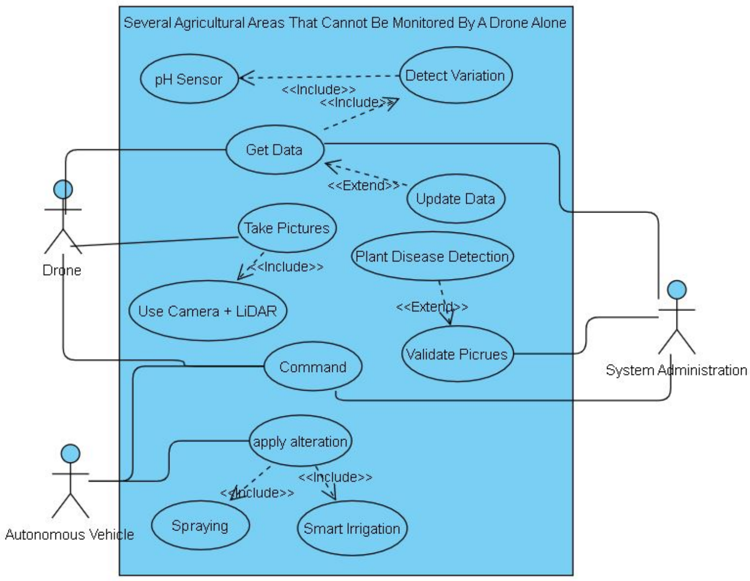


Fig. A.13: Use Case Diagram for Several Agricultural Areas That Cannot be Monitored By Drone Alone