Visualising the work protocol

Hardware required-

Option1: We can use MQ1 for scanning the bed. It has embedded system attached that can store the images with proper tags for data processing

Option2: PE system can be used for data collection purpose running it on forecast mode. It will only use its camera and the processor for sorting the images with tags for analysis. At the moment frame model is ready, the processor needs to be decided depending on fetchable sensor data

Data collection routine

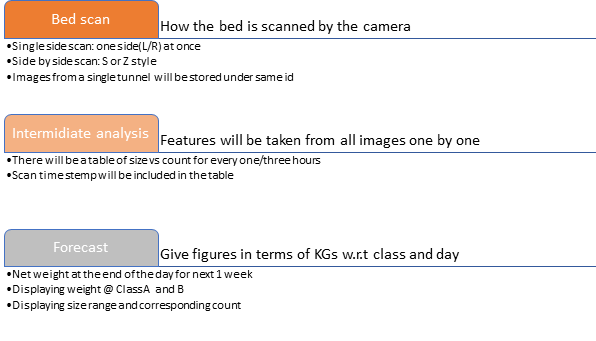
Scans will be done either single side wise e.g. left side of the shelf will be scanned first following the other side or side by side e.g. a small portion of shelf to be covered in left to right or vice versa pattern

It may seem as U scan or S/Z scan depending on the move of the camera

Once a full shelf is scanned, all corresponding images will be stored naming like T{tag}S{tag}\_hh\_num.jpg. T is tunnel, S is shelf, hh is the hour when scan took place, and num being the id of the image. These data will be stored under day folder naming like Scan\_dd\_mm, that will contain a specific day’s scanned data.

Data processing routine

Starting with forecast, images from a particular tunnel will be considered at first. From individual images, size range and count will be generated for the whole tunnel taking all images under processing. As described earlier, with range and count, firstly, net weight of yield from that tunnel can be derived; secondly, every three-hour basis net production can be forecasted; Thirdly, current size of mushroom to harvest would be reconned; last but not least, with image processing with class features, net production would be displayed divided into Class A and B



Visualising the raw data and forecast model

As described previously, data storing will be done using the name of the images with specific tags and parent directory, giving least data management issues.

Q) What data is required to feed into the forecast model?

First of all, a whole bed scan will be performed to get data for that bed. Refer to the Data processing routine where whole forecast process is divided into three sub processes.

Once a full bed data is fetched, by segmentation algorithm, available size ranges(generations) corresponding to their counts will be derived. With size and count approximate weight can be computed. For example, if 20-25mm mushrooms are 200 in count that means 200\*1.5g=300g would be the forecasted weight of 20-25 mm mushroom from that bed, given in average a mushroom within that range weighs 1.5g.Now, let's say there are 20 shelves in a tunnel, therefore summing up all forecasted weights for 20-25mm mushrooms would lead to the forecast of 20-25mm for the tunnel itself.

Below is a sample scenario where every 3 hours of scanned data is stated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 am | 10 am | 1 pm | 4pm | so on |
| 20-25mm 300kg  30-35mm 350kg  35-45mm 600kg  ... | 20-25mm 200kg  30-35mm 380kg  35-45mm 500kg  ... | 20-25mm 330kg  30-35mm 350kg  35-45mm 600kg  ... | 20-25mm 150kg  30-35mm 350kg  35-45mm 600kg  ... | ... |

During data collection compost type needs to be considered as synthetic compost produces more yield than conventional compost. This means growth variation may be different. Therefore, it would be safe to make a model with conventional compost data at first.

Forecast model

Model architecture

Test will begin with existing statistical models like Gaussian Process Regression(GPR), Gradient boosting, and ARIMA (auto regressiveintegrated moving average).

Size range under consideration (this is made considering cap and flat and other variety of mushrooms either)

|  |
| --- |
| 20-25mm |
| 25-35mm |
| 35-45mm |
| 45-55mm |
| 55-65mm |
| 65+mm |

**Option 1**

Model built on Weight: -

This model will take time series data as mentioned in the table under Raw data visualisation section. It will take every 3 hours data in terms of weight starting from 7 am to 7 pm and try to understand variation of weights in time for each size range (20-25mm, 25-35 mm etc.)

To train the model, given it forecasts data in next 3 hours a huge number of batch will be provided with input and output. Once it is trained it will be able to give data in next 3 hours and this is how we can get a forecast for a whole day and a week with a little trick.

Mathematics behind weight calculation:

Weight=size\*count (refer to Raw data visualisation section)

Single bed weight= Sum of weights corresponding to all sort of ranges on bed

Net weight/tunnel = Sum of all bed weights in a single tunnel

|  |  |  |
| --- | --- | --- |
| Input | Output | Input and output are in kgs(weight) |
| 7 am | 10 am |
| 10 am | 1 pm |
| 1 pm | 4 pm |
| 7pm | 7 pm |

While scanning mushrooms will be classified into Class A and B. at the end of the scan a percentage of class A and B will be derived. In forecast similar percentage will reflect the grading.

**Option 2**

Model built on count:-

Similar option 1, it will take time series data but instead of weight (product of size range multiplied by count) it considers counts. In other word, it will track how much a single mushroom grown over time let us say in every 3 hours. This will lead to count change w.r.t specific size range (20-25mm, 25-35mm etc) over that time. This count w.r.t size ranges will be the input to the model. As an output the model will give counts in 3 hours ahead, corresponding to different size ranges. In this way the model will try comprehending the change of mushrooms over time and forecast upon training finished.

To calculate net weight again, counts corresponding to each range will be multiplied and at the end added all together to give rise net weight for a shelf/bed. Once all forecast weights per shelf are added, net weight per tunnel can be forecasted. Refer to the mathematics behind weight calculation section above.

Ground truth:

In terms of evaluation or ground truthing the forecasting model, future scan will be ground truth of previous scan. For example forecast over scan at 7 am will be calibrated with scan at 10 am for ground truthing e.g. t+3 will be ground truth of t when t is the time of scan.

|  |  |  |
| --- | --- | --- |
| Forecast Data | Ground truth | This table will carry on the based on the scan frequency over time |
| 7 am | 10 am |
| 10 am | 1 pm |
| 1 pm | 4 pm |
| 4 pm | 7 pm |

To train the forecasting model, given it predicts in 6 and 12 hours, data needs to be given like

|  |  |  |
| --- | --- | --- |
| Input | Output | 3 hours interval for 12 hours(day shift) |
| 7 am | 1pm,7pm |
| 10 am | 4pm,10 pm |
| 1 pm | 7pm, 1am |
| 7pm | 1 am, 7 am |

**Option 3**

Model Built on Time series growth:

In this process during scanning each sizes will be taken care of. If scanning happens every hour, individual size growths per hour will be recorded. For example, at 7 am 20mm to 65mm sizes will be recorded and tagged under 7 am. Next hour another scan will be performed and growth in one hour will be recorded. This will repeat every hour starting from 7 in the morning to 7 in the evening.

Once scan data is collected this growth data consisting of hourly growth of particular sizes will be fed to ARIMA model and then based on the model parameters forecast will be generated for next 6 or 12 hours. Once the forecast is done, comprehending the weight per sizes will be pretty straight forward. Weight can be calculated using the formula- weight= count\* size factor. Here size factor is the approximate weight of a particular size corresponds to. Net weight can be fetched by summing all the weights with respect to each sizes. This is how net weight from a tunnel can be estimated.

Use of the model:

The trained model will be able to give forecast in terms of growth (mm) in distant future. With this help one can apprehend how much a 20mm mushroom can grow in 6, 12, 24 or so on hours. And based on that table growth of available sizes in a specific time can be forecasted.

Everyday full bed scan will be performed at 7 am. Sizes with relevant counts will be recorded. The model will give 6/12 hours growth ahead of time and based on that weight can be calculated and eventually the net weight for a tunnel.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Forecast data | | |
| Size | 6 hrs | 12hrs | 24hrs |
| 20mm | 26mm | 32mm | 38mm |
| 21mm | 29mm | 34mm | 40mm |
| 60mm | 65mm | 85mm | 110mm |
| Xmm | X’mm | X’’mm | X’’’mm |

Model drawback and inferences :[Model experiemts and Growth rate](https://universityoflincoln-my.sharepoint.com/:w:/g/personal/akundu_lincoln_ac_uk/EWhzdNfqMepMu6lr3xnmIwsBKlEtj-m3KBb7f2K9S_UBYQ?e=XWfcKN)

Requirements:

For destructive forecast methodology, every hour scan will give 3 times more data samples than that of every 3 hours scan. If the system is decided for destructive way of forecasting, every hour from 7 am to 6 pm might help to get more data to train the model. Again, every hour scan data needs to be different otherwise we can go for every two hours scan. Also, one thing needs to be decided for training the model is the period of forecast e.g. the system will forecast every 6 hours/12 hours/24 hours at primary level and then with that data weekly forecast can be derived.

If the forecast system gives forecast on 24-hour basis, it will lead to a weekly basis forecast (primary target) and even bi-weekly or monthly. Harvest is decided based on daily forecast so our target should focus on a 24-hours forecast or furthest 12 hours forecast. With this approach names re-generative forecast, the system will be robust with current time growth and adaptable

Re generative forecast (Forecast routine)

How forecast routine will take place everyday time to time. Below is an example for visualisation how forecast data will be updated in different part of a day with an interval of 6 hours. This will help team leaders understand mushroom growth in hours aiding them go more precise with size and placing pickers in individual tunnels. Farm includes pickers as a part of forecast. For automated harvesting this part can be handled by the harvester. As there is a direct corelation between no of pickers with pickers quality and weight/tunnel (sometime with flush: experienced ones are placed in 1st flush whereas new and 3rd flush allocates low experienced pickers) this can be easily formulated in need.