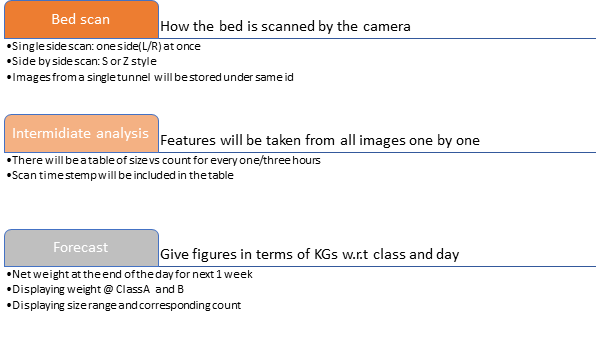
Data insight in Forecast modelling

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



Raw data visualisation

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 taken when the input feature is weight

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  ... | ... |

If count is the input feature, the sample above remains same but there will be counts in number instead of weights in kgs.

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

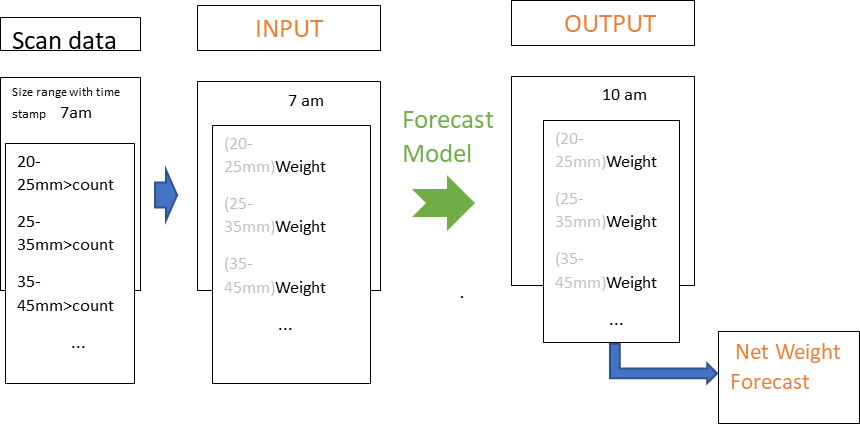
Model architecture

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

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 batches 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) e.g.

w = Πs-range\*count ; count=number of mushrooms corresponding to s-range=size range

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

wshalf =∑ws-range ; s-range ∈ 20-25mm, 25-35mm, 35-45mm and so on

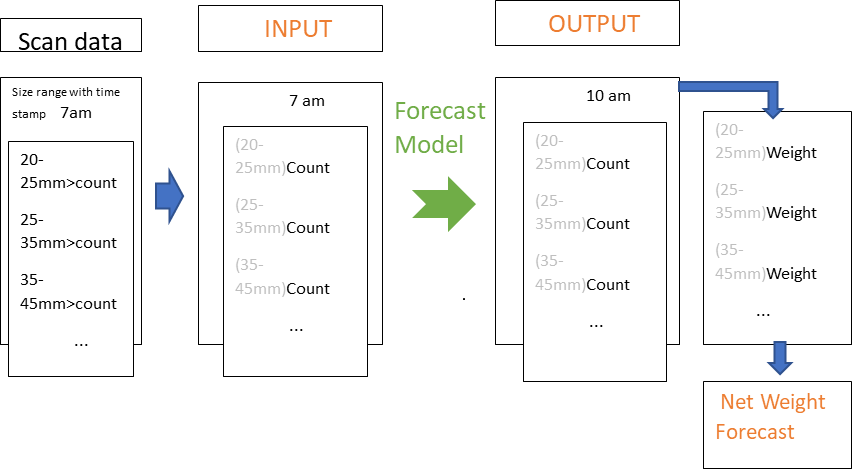
Net weight/tunnel = Sum of all bed weights e.g. and so on in a single tunnel e.g.

wnet =∑wshelf ; shelf ∈ shelf 1, shelf 2 ,..., shelf 20

|  |  |  |
| --- | --- | --- |
| 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 distinct 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.

**Option 3**

As a later option, a mixer model can be implemented upon satisfactory test of both options. In simple work let’s say the model built on count will be comparted with the model built on weight. A mixer module will be made for this purpose. The best outcome will be chosen by the mixer and that will be shown as forecast.

Forecast process in subprocesses: -

During training data collection,

1. Scanning a whole bed every 3 hours will be performed. Each scan data will be tagged with time like 7am scan, 10am scan and so on.
2. Counts per size range (mentioned in model architecture) will be derived by data processing over the scan data. This will result in weight for each size range and finally net yield for a tunnel
3. Either weight or count corresponding to each specified size range will be fed to the model as feature inputs expecting same parameter e.g. weight or count as output forecast in 3 hours ahead of the time of input feature
4. The working principal of the model alongside sample data(input features and expected output) is described in model architecture

Once forecast model is ready,

1. There will happen 2 scans in 24 hours, one at 6 am and the other at 12 pm. This scan will help the model forecast very precisely on daily basis with internal adaptation.
2. Once scan data is processed, mushroom counts corresponding to distinct sizes will be tubulised for every tunnel. This data will be tagged with time(6am/12pm). As mentioned above two scans will be performed a day.
3. The forecast model will take count or weight for every size range and these will act as features input in the model
4. Based on its training and learning the model will forecast weight (KGs) in 6 hours and subsequently for a complete day. Based on daily data it will forecast for the rest of the harvesting period
5. As mentioned earlier, while scanning grading percentage (Class A and B) would be calculated and that percentage with a little tweak will be used for forecast grading

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 it 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

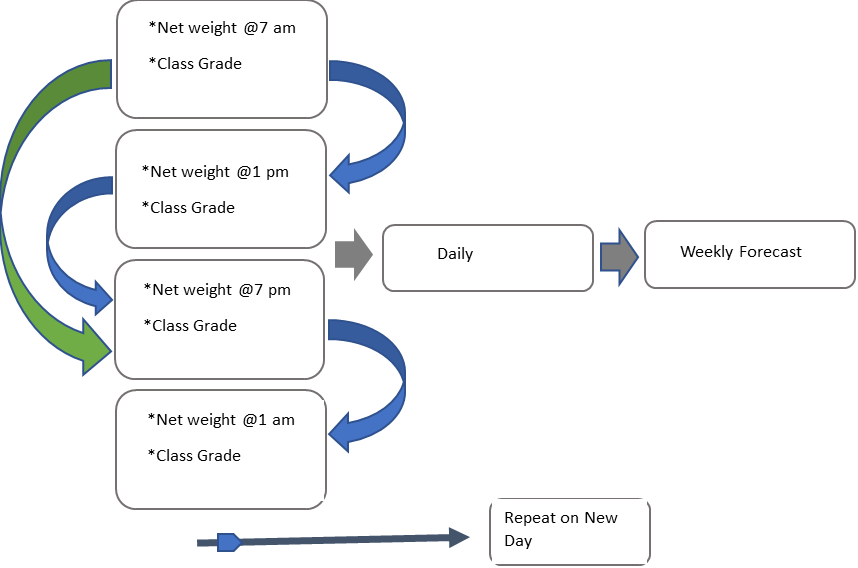
|  |  |  |
| --- | --- | --- |
| Input | Output | 3 hours interval for 12 hours (day shift) |
| 7 am | 1pm,7pm |
| 10 am | 4pm,10 pm |
| 1 pm | 7pm, 1am |
| 7pm | 1 am, 7 am |

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.



Bridging sensor and forecast data

The sensor data can be divided into two categories: compost sensing, and ambience sensing

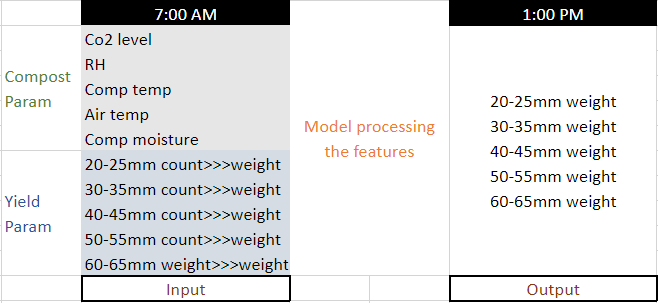
Compost sensing: Sensing directly linked or attached to compost is compost sensing e.g. pH, compost temperature and moisture sensing. These three sensors are worthy to be mounted on the harvester.

Ambience sensing: RH sensor and Air temperature sensing comes under ambience sensing due to the fact that RH and Air have nothing to do with compost

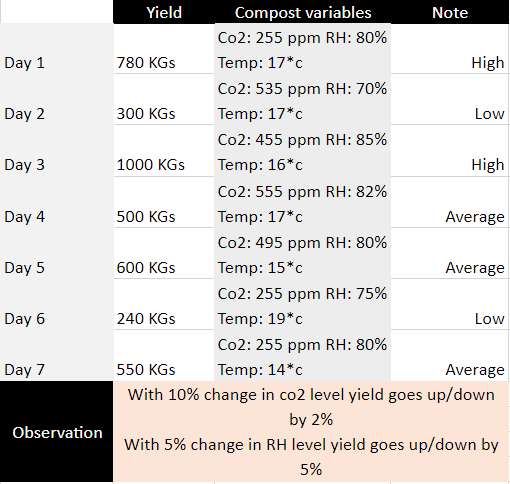
Forecast model will forecast based on scan data concurrently there will be a set of sensor data tagged per forecast. While scanning, both compost and ambience data will be collected.

There are two possibilities, the data can be assembled with the scan data-

1. Sensor data can be input to the model as features alongside mushroom features e.g. weight/count and the forecast model can be built with the features all together. The issue with this approach is that if the sensor data is not dynamic there may happen feature cross disparity. It happens when relevant features are not fed to a model.



1. The forecast model will be built on mushroom features as discussed above. There might be an environment *proportional factor* that can be derived what influences yield while doing data analysis per scan given while scanning environment data will be tagged so that this can be compared with different scan data, and it can be possible to find out the proportional factor. Once the proportional factor is derived forecast data will be multiplied with the factor to finalise the forecast.



Proportional factor is a factor is a number that bridges sensor data variation with yield data variation. For example, with a change of 5% in environment data yield goes 7% up or down

At the end, Environment data is worthy to record per scan so that it can be compared with yield and in time. This would help do data analysis to find how the environment data is liked to mushroom yield.

1. If the model takes mushroom growth time series data e.g. checking how much a mushroom is growing ever hour and doing forecast in 6 hours ahead based on that growth(mm), it sounds even tricker to ensemble the sensor data with the growth. It might be worthy to have a deep look at the sensor data over a week how it changes over time (every12 hours and every day). If it finds out to be changing very frequently let’s. say every 6 or 12 hours it might be worthy to consider the major sensor factors like compost and air moisture over others and put those readings as features. Otherwise, if the sensor reading seems stationary over long period let’s say it changes on day-to-day basis, then it might be worthy to filter the major sensor factors and record corresponding daily mean sensor readings. These readings will be concatenated with shelf yield per tunnel (KGs). In this way we can track normal yield with corresponding sensor reading and so for worse and better yield. This proportionate data will be used to get précised forecast when environment changes.

Required experiment:

In a test confined environment, sensor data needs to be recorded with weight/growth and getting a proportionate factor wrt normal environment will help get forecast when sensor data changes as described above.

How Farm is using environmental data

The farm receives a compost datasheet where all environmental variables are set and mentioned. Due to mushroom growth and spontaneous reaction compost and environment go through changes. Farm tries to bring the changed parameters back to the preset level by adjusting other relevant parameters. For example, when co2 level goes high, air is brought into the tunnel to bring the co2 level back and vice versa. Same for air and compost temperature.

The farm is using the environment data-

1. Make compost (temperature, wetness) and ambience (air temperature, co2 level, airy, humidity) parameters steady and matched with the preset values.
2. Understanding if any trend exists between compost parameters and Yield.

What can be done from the harvester side

Using the sensor data the following can be implemented-

1. Compost and ambience monitoring time to time, even more precisely and accurately
2. Making compost and environment steady by automatically adjusting the contributing parameters in need
3. Determining possible pattern linked in between compost variables and yield as mentioned in possibilities above