

# Robotic Crop Disease Monitoring Using Neural Network-based Prediction and Weighted Path Planning

Jacob Sutton, Ayan Dutta, O. Patrick Kreidl, Ladislau Bölöni, Swapnoneel Roy

**Abstract**—Disease control is paramount in modern agriculture to ensure optimal yield. Monitoring the spread of crop diseases is crucial for effective control measures. Traditional methods involve uniform pesticide spraying across entire fields, which can be inefficient and environmentally harmful. In this paper, we propose an intelligent solution employing mobile robots equipped with predictive AI techniques for disease monitoring and targeted intervention. These robots strategically visit select locations within the field, guided by a convolutional and recurrent neural network model trained on limited data to predict disease spread. We introduce a novel weighted path planning algorithm to optimize robot movement within the field considering disease risk and battery constraints. Our approach is implemented in the WaterBerry benchmark, an open-source platform for agricultural robotics. Experimental results demonstrate the efficacy of our technique, showcasing improved prediction accuracy and operational efficiency compared to baseline methods.

## I. INTRODUCTION

Agriculture plays a vital role in sustaining global food security and economic stability. However, agricultural productivity is constantly threatened by various factors, with crop diseases posing a significant challenge. Crop diseases such as Tomato Yellow Leaf Curl Virus (TYLCV), not only result in yield losses but also incur substantial economic and environmental costs due to the indiscriminate use of pesticides and fungicides [1], [2], [3]. Traditional disease management practices often involve uniform spraying of agrochemicals across entire fields, regardless of disease presence or intensity. While this approach may provide some level of protection, it is often inefficient, costly, and environmentally detrimental. Moreover, the indiscriminate use of agrochemicals can lead to the development of pesticide-resistant strains of pathogens and harmful effects on non-target organisms [4].

To address these challenges, there is a growing interest in adopting precision agriculture techniques that leverage advanced technologies such as AI-enabled robotics and sensing networks for targeted disease management. In this context, mobile robots equipped with sensors and AI algorithms present a promising solution for real-time disease monitoring and intervention thereafter [5], [6].

In this paper, we propose a novel approach for robotics-based disease spread monitoring in agricultural lands that

This work is supported in part by NSF CPS Grants #1932300 and #1931767.

J. Sutton, A. Dutta, O.P. Kreidl, and S. Roy are with the University of North Florida, USA. Emails: {n01379084, a.dutta, patrick.kreidl, s.roy}@unf.edu

L. Bölöni is with the University of Central Florida, USA. Email: ladislau.boloni@ucf.edu

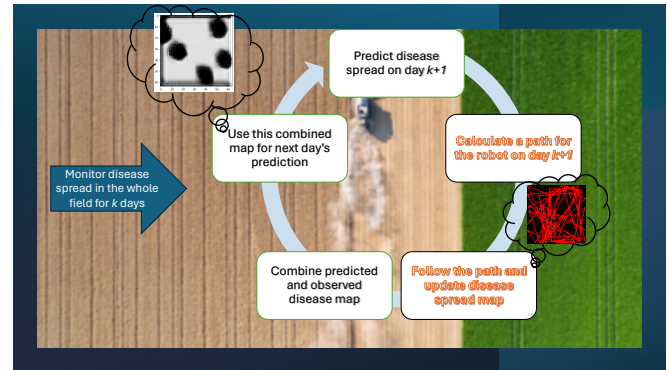


Fig. 1. The overall architecture of the proposed disease spread monitoring framework. An example of a day's path calculated by the robot is also shown with red lines. The corresponding day's combined predicted and observed disease spread (black cells) in various parts of the field is shown (top left).

uses a deep neural network-based prediction technique along with a novel weighted random path planning. More specifically, our approach involves two key components: (1) a predictive AI model based on convolutional and recurrent neural networks (CNN and RNN) for disease spread prediction, and (2) a novel weighted path planning algorithm that considers both disease spread risk prediction outputs by the network and robot battery constraints for field-wide robot inspection for first-hand verification of the disease spread. We implement our proposed framework within the open-source WaterBerry benchmark [7], providing a standardized platform for evaluating and comparing different disease monitoring and control strategies. Simulation results show that our proposed technique outperforms two baselines in terms of building close-to-reality disease spread maps. Such high-quality disease spread mapping has significant practical implications as the farmers' targeted intervention plans will rely on this.

## II. RELATED WORK

Disease mapping and its spread monitoring is an important task for farmers that can save crop yields from getting destroyed by fast-spreading diseases such as the Tomato yellow leaf curl virus. In [8], the authors proposed a novel statistical model that incorporates both spatial and temporal correlations to enhance the accuracy of disease mapping over time and geographical areas. By employing autoregressive techniques, the model effectively captures the dynamics of disease spread, facilitating more precise public health surveillance and intervention planning. Su et al. [6] examine the use of unmanned aerial vehicles (UAVs) equipped with

multispectral cameras to detect and monitor the spread of yellow rust in wheat crops over time, offering a high-resolution, cost-effective tool for precision agriculture. In recent years, a plethora of novel approaches have been proposed in the literature that use machine learning techniques for such disease mapping. Shinde and Kulkarni [4] systematically examine the integration of Internet of Things (IoT) technologies and machine learning algorithms to predict and manage crop diseases. It highlights advances, challenges, and prospects in this interdisciplinary field, demonstrating the potential for real-time, data-driven agricultural decision-making. The authors in [3] find that machine learning models outperform traditional process-based models in accuracy and efficiency for predicting the occurrence and spread of rice blast disease. In [9], the authors explore the use of RNNs to dynamically predict crop disease severity over space and time, aiding in more effective agricultural emergency responses. [10] focuses on developing and applying anomaly detection techniques to identify and predict unusual patterns in crop health data, aiming to forecast the risk of disease outbreaks. Detection of wheat disease using multi-spectral remote sensing techniques is studied in [11]. Ghosh et al. [12] investigate the effectiveness of using Sentinel-2 satellite imagery to monitor crop health and development across different spatial and temporal dimensions. [13] proposes using machine learning techniques (namely neural networks, KNN, Random Forest, and Linear Regression) to forecast the occurrence of pests and diseases affecting *Coffea arabica*, aiding in proactive pest management strategies for coffee cultivation. In [1], the authors proposed a system that combines connected sensors enabled with machine learning techniques to collect and analyze real-time data for early prediction and management of diseases in mango crops. In our work, we employ a convolutional recurrent network proposed in [14], which introduces a machine learning model that combines CNNs and LSTM units to improve the short-term forecasting of precipitation, demonstrating significant advancements in weather prediction accuracy.

### III. PROBLEM SETUP

We have a mobile robot  $r$  with a sensor attached to it to sense the condition of the crop. For example, an unmanned aerial vehicle with a camera on it looking down.  $r$  is also equipped with a GPS to locate itself as well as for navigation. Due to onboard battery limitation, the robot can only travel for  $b$  unit distance before it needs to go back to the central station for recharging. The agricultural field is a polygon  $F$  which is discretized into  $C$  cells. Let  $C_d^i$  and  $C_h^i$  denote the set of diseased and healthy crop cells on the  $i$ -th day. Thus,  $C = C_d^i \cup C_h^i$ . Let  $O^i$  denote the cells (along with their disease states) on the robot's  $i$ -th day's path  $W^i$ .

Note that our study does not incorporate disease management or intervention techniques by incorporating pesticide-spreading schedules into the proposed algorithms. Because of this, we assume that if a cell is part of the  $C_d^i$  set, it will remain in this set till the end of the process. In a scenario, where regular pesticide spraying is happening, a subset  $c_d \subseteq$

$C_d^i$  can move to  $C_h^{i+1}$ . On the other hand, our proposed framework already assumes that a subset  $c_h \subseteq C_h^i$  can move to  $C_d^{i+1}$  due to the nature of the disease spread. This is also true for the TYLC virus modeled in the simulations [7], [2].

We assume that the cell that the robot is in currently, only that cell's information (e.g., a picture) can be collected by the robot. One can think of a scenario where the robot takes a picture of multiple cells surrounding it at the same time (i.e., assuming the conical vision of the camera system) but that will not impact our proposed framework's foundations instead more information will be passed to it for potentially better-quality disease spread prediction.

**Disease Spread Formulation.** In this paper, we use the Waterberry Farms benchmark presented in [7] for such agricultural simulations. The disease spread formulation is used as is from this benchmark and discussed briefly in this paper next.

The disease spread formulation uses the SIRI model of disease propagation adapted to a 2D environment. Every grid cell in the environment is assumed to be in one of the possible states of Susceptible, Infected, Recovered, and Immune. To illustrate how this works, let us consider the Tomato Yellow Leaf Curl Virus (TYLCV), a disease of tomato plants that leads to total crop loss. Healthy tomato plants are in the Susceptible state, while diseased plants are in an Infected state. The likelihood for a tomato plant to become infected during a particular timeslot is affected by (a) an underlying, relatively low static probability not affected by nearby infected plants and (b) a probability that is dependent on the proximity and number of nearby infected plants. Overall, the probability of infection can be calculated by a convolution with a kernel which describes the decreasing likelihood of infection with the plants. The plants remain infected for a certain amount of timeslots and then transition to the Recovered state. In the case of the TYLCV, this means a dead plant, which is not a source of further infection. For this scenario, Immune cells are cells outside the considered area, bodies of water, or areas planted with plants that are not infected by TYLCV (in the case of the Waterberry Farms model, these are areas planted with strawberries).

**Objective.** The objective of the proposed framework is to use a robot to collect disease spread information from the crop cells by traveling at most  $b$  distance on the  $i$ -th day such that the error between the combined predicted and observed map (denoted by  $Y^{i'}$ ) and the ground truth disease spread map (denoted by  $Y^i$ ) is minimized. More formally, find a path  $W^{i*}$  containing the cells  $O^i$  such that

$$W^{i*} = \arg \min_W (Y^i - Y^{i'})^2 \quad (1)$$

How specifically the combination of the predicted disease map and the observed disease cells happens, is discussed in more detail in the next section.

#### IV. WEIGHTED PATH PLANNING WITH NEURAL NETWORK-BASED FORECASTING FOR DISEASE MONITORING

##### A. CNN-LSTM-based Prediction

In recent years, neural network architecture models became the standard in developing predictive models for sequences such as time series. The prediction models can be roughly classified into models that accumulate incoming data into a memory structure versus models that retain the input sequence and use attention to focus on different areas of it. LSTMs, GRU, and other state-space models are examples of the former class, while transformer architectures are examples of the latter. For our problem, we need to solve the disease spread prediction problem on a two-dimensional area, which requires both the input data and prediction output to be a 2D array, in which spatial relations are significant. The neural network architecture we use is the Convolutional LSTM model [14], an architecture originally proposed for predicting precipitation over a geographic area, which combines the temporal prediction model of LSTM [15] with the spatial reasoning capabilities of CNNs.

This model modifies the vector inputs of the LSTM model to be of the form inputs  $\mathcal{X}_t$ , outputs  $\mathcal{C}_t$ , hidden states  $\mathcal{H}_t$  in the form of tensors with the last two dimensions being the spatial dimensions of a geographical area. With this model, the equations that describe the behavior of the ConvLSTM model are:

$$\begin{aligned} i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\ \mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t) \end{aligned}$$

where  $*$  and  $\circ$  denote the convolution operator and the Hadamard product respectively, and  $W$  and  $b$  are learnable weights of the model.

##### B. Combining Prediction and Weighted Random Planning

This section proposes a novel algorithm motivated by the findings in [16]. The pseudocode for this is presented in Algorithm 1. Let  $O^{1:k}$  denote the cells visited by the robot (or, human) and their disease states. Only the  $k$ -th day's disease state is stored, not the history. This is done as disease spread might not start on the 0-th day itself. We assume that the disease is first noticed on the  $k'$ -th day where  $0 \leq k' \leq k$ . We do the following for the next days past the  $k$ -th day. The prediction of the diseased cells on the  $k+1$ -th day is made based on the observations until the  $k$ -th day. The CNN-LSTM-based forecasting technique described earlier outputs a 2D array of size  $C$  where the array contains numbers in  $[0, 1]$ . Any cell with a value below 0.5 is predicted to be diagnosed as diseased. Let  $C_{dp}^{k+1}$  denote the predicted diseased cells on day  $k+1$ .

Based on the prediction, a path  $\mathbf{W}^{k+1}$  is calculated for the  $k+1$ -th day. The robot visits the cells on path  $\mathbf{W}^{k+1}$  on day  $k+1$ . After this, the observation map for that day,  $O^{k+1}$ , is updated. Next, we merge the observed disease cells according

---

#### Algorithm 1: Disease spread prediction and path planning for autonomous monitoring

---

**Output:** A disease spread map  $C$  after  $k+n$  days.

- 1  $O^{1:k} \leftarrow$  Observe all cells for the first  $k$  days and store their disease states;
  - 2  $\mathbf{P}$ : A probability distribution that determines the weight of choosing a cell;
  - 3  $b = \text{velocity} \cdot \text{time}$
  - 4  $D \leftarrow$  Divide area into a grid of size  $h_{cells} \times v_{cells}$
  - 5  $i = k$ ;
  - 6 **for** each day until  $i < n$  **do**
  - 7    $C_{dp}^{i+1} \leftarrow$  Predict the disease cells on day  $(i+1)$  using the CNN-LSTM technique [14];
  - 8    $\mathbf{W}^{i+1} \leftarrow$  Calculate a path for the robot to follow on day  $(i+1)$  (Algo. 2);
  - 9    $O^{i+1} \leftarrow$  Update the observations after visiting the cells in  $\mathbf{W}^{i+1}$  on day  $(i+1)$ ;
  - 10    $C_{dm}^{i+1} \leftarrow$  Merge the observations made by the robot,  $O^{i+1}$ , and the predicted disease cells  $C_{dp}^{i+1}$ ;
  - 11   Include  $C_{dm}^{i+1}$  to the test set of the CNN-LSTM prediction;
  - 12    $i = i + 1$ ;
  - 13 **return** the final merged predicted and observed map;
- 

to the observations of that day  $O^{k+1}$  and the predicted disease cells  $C_{dp}^{k+1}$  such that if any cell  $c \in C \setminus C_{dp}^{k+1}$  is observed to be diseased by the robot, we insert  $c$  to  $C_d^{k+1}$ . On the other hand, if a cell  $c'$  was forecasted to be diseased but was found to be healthy when the robot visited it, then we add  $c'$  to  $C_h^{k+1}$  while removing it from  $C_{dp}^{k+1}$ . Following this, the merged predicted and observed diseased cells are kept in a set  $C_{dm}^{k+1}$ . Note that the robot might not have the chance to visit all the predicted disease cells. Therefore, the cells forecasted to be diseased but not verified by the robot, are assumed to remain diseased for the next round of forecasting (until not visited and verified by the robot). After  $n$ th day this cycle stops and the merged final map of predicted diseased and observed diseased cells along with the healthy ones are returned.

**Path Planning (Algo. 2).** The robot starts the exploration from a fixed cell  $c_s \in C$ . This can be the farmer's control station. The budget for each day  $b$  is decided by the robot's velocity and how much time it can operate with a full charge while using its sensors (typically 30 mins. for a drone). The environment  $F$  is divided into  $|D|$  sub-regions, where each division  $d \in D$  has  $h_{cells} \times v_{cells}$  cells. The rationale for dividing the area into sub-regions is for the robot to visit various parts of the agricultural field to detect new outbreaks of disease or to monitor the spread.

The robot first randomly chooses a sub-region  $d$  to visit. Next, it chooses a cell  $w \in C$  from  $d$  using a user-defined probability distribution  $\mathbf{P}$ , which determines the probability of a cell being selected based on its disease state. The three

**Algorithm 2:** Path planning on day  $j$ 


---

**Input :** Start cell ( $c_s$ ),  $C_{dp}^j$ ,  $\mathbf{P}$ ,  $b$ ,  $D$   
**Output:**  $path$

```

1  $path = [c_s]$ 
2 repeat
3   Choose region  $d \in D$  as the next region to
   explore;
4   Choose waypoint  $w$  from  $d$  using the probability
   distribution  $\mathbf{P}$ ;
5   if Going to  $w$  will be over-budget then
6     Append the control station to the end of  $path$ ;
7     return  $path$ ;
8   else
9     Append  $w$  to end of  $path$ ;
10 until  $length(path) < b$ ;
11 return  $path$ ;
```

---

states of a cell that are being considered in  $\mathbf{P}$  are

- Visited, healthy: a cell that was visited by the robot on a prior day and was found to be healthy.
- unvisited: never visited, i.e., not part of  $O^{1:j-1}$ .
- forecasted, unvisited: a cell predicted to be diseased, yet not confirmed by the robot through observation.

Such a probability distribution allows the robot to inspect 1) unseen locations to find out whether there is a new disease outbreak location or not, 2) a visited healthy cell has recently become diseased or not, and finally, 3) to verify the prediction by the forecasting model for better modeling of the disease spread in the future. Such a cell  $w$  is repeatedly added to the robot's  $path$  for the  $j$ -th day.

The robot keeps its camera on for disease monitoring between two waypoints in  $\mathbf{W}^i$ . Therefore, all the cells being passed through by the robot (calculated using Bresenham's line algorithm) are assumed to be observed. If the robot finds out that going from the current cell to  $w$  and then from  $w$  to  $c_s$  combined with the already-incurred path cost to get to the current cell goes above budget  $b$ , it adds the control station to the end of the  $path$  instead of  $w$  and returns the  $path$ .

TABLE I

LIST OF PARAMETERS USED IN OUR EXPERIMENTS AND THEIR VALUES.

| Parameter                         | Value                                             |
|-----------------------------------|---------------------------------------------------|
| Environment size                  | $64 \times 64$                                    |
| Budget ( $b$ )                    | $\{0.1, 0.2, 0.4, 0.6\} \times 64 \times 64$      |
| Training data size                | 1000 (generated within Water-Berry framework [7]) |
| Test cases                        | 100                                               |
| Distribution $\mathbf{P}$         | $[0.1, 0.6, 0.3]$                                 |
| $h_{cell}$                        | 32                                                |
| $v_{cell}$                        | 32                                                |
| Number of regions                 | 4                                                 |
| Complete observation days ( $k$ ) | 10                                                |

## V. EXPERIMENTS AND RESULTS

The proposed disease spread monitoring framework is implemented in Python. The simulated agricultural environment

is a  $64 \times 64$  grid. Table I lists the parameters and values used in the experiments. Any cell with a probability of more than 50% of being in the disease *class* predicted by the prediction technique is classified as diseased and added to the set of predicted disease cells for that day, i.e.,  $C_{dp}^i$ . The start cell  $c_s$  is (0,0). We have used the Euclidean distance as our distance metric. The prediction results for 20 days in these test cases after the first  $k = 10$  complete observation days are presented next.

We have compared our proposed technique against two relevant baselines as follows.

- **No Path.** In this baseline, we use the prediction only from the CNN-LSTM-based forecasting technique. Day  $i$ 's predictions are assumed to be completely accurate and used as is for the disease spread predictions for day  $i + 1$ . No actual robot movement/observation is taken into account, i.e.,  $\mathbf{W}^i = \emptyset$  in line 8 of Algo. 1. This goes on up to the  $n$ -th day.
- **Pure Random.** In this baseline, the robot follows a  $b$ -length path every day for up to  $n$  days. In this random planning strategy, the robot chooses one of its four neighboring cells to be the next waypoint  $w$ . After it exhausts its budget, it returns to the control station. As the disease spread can start at different locations of the environment and can progress at different rates, the random baseline carries merit.

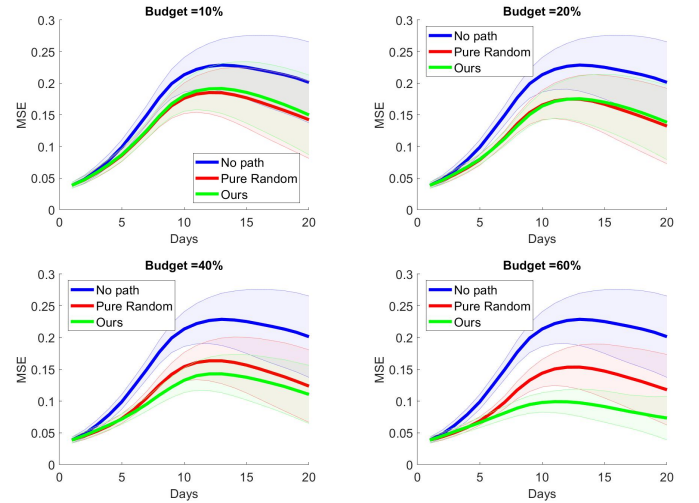


Fig. 2. Accuracy (in terms of MSE) of building the disease spread map with various techniques for different budget amounts.

## A. Results

The main objective of the proposed framework is to create high-quality, i.e., close-to-reality predicted disease maps for the farmers so that necessary steps can be taken to stop the spread of the disease. To validate this, we use Mean Square Error (MSE) between the ground truth disease map and the combined map of predicted and observed disease states. The results are presented in Fig. 2. In all the budget cases, we see that the 'no path' baseline performs the worst in terms of predicted disease map accuracy. This is expected as in this

setting, the robot does not visit any location to observe the real state of the disease spread, instead relies completely on the neural network model-based disease forecasting. On the other hand, our proposed framework performs comparably with the random approach. However, with the increased budget amount, our proposed model outperforms the random baseline as well. Unlike the random model, our proposed framework lets the robot 1) span the entire environment and 2) plan its path while incorporating the predictions and observations from the prior day. With budget = 60% of the environment, our proposed technique achieves an MSE of 0.07 while the random and the no path baselines achieve MSEs of 0.12 and 0.20 respectively on the last day. The initial prediction MSE is low as up to that point all the cells in the environment were under observation for  $k = 10$  days. Also, we notice that for all the tested approaches, the MSEs rise to  $\sim$  day 10 and then decrease. We found out that till  $\sim$  day 10, new areas are getting affected by the disease and are more unpredictable. However, after that tip-off point, such unpredictability decreases all the variants perform increasingly better in terms of MSE. Even within such a complex disease-spreading crop field, our proposed framework incurs the highest MSE of 0.09 with  $b = 60\%$  that is lower than the final MSEs achieved by both ‘Pure Random’ and ‘No Path’ baselines.

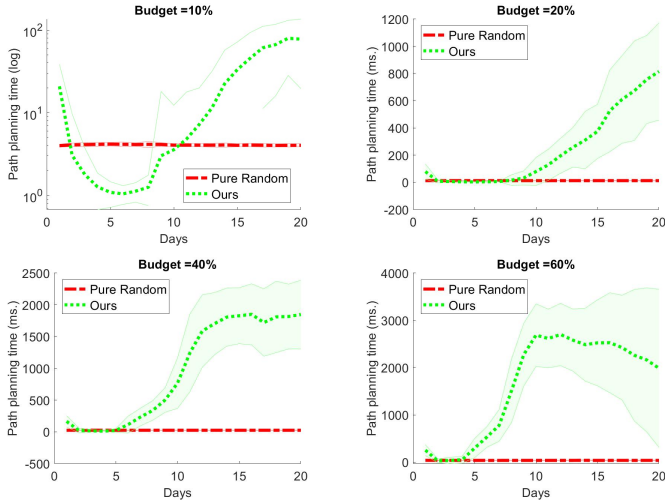


Fig. 3. Run time for planning the path for the  $(i + 1)$ -th day.

Next, we are interested in investigating the execution time metric. Prediction time (using the trained neural network model) as well as path planning time (using Algorithm 1) are the primary components of the time metric. First, we investigate the path planning time. Given that we have presented a sophisticated informed search-based path planning strategy compared to the ‘Pure Random’ baseline, it incurs a higher path planning time cost. Note that this is the cost to calculate the waypoints every day for day  $k + 1$  to day  $n$ . The results are presented in Fig. 3. Although higher than the random baseline, our planning time is still nominal – the maximum being a mere 2.70 sec. When combined with the prediction time, the total execution time is higher for our

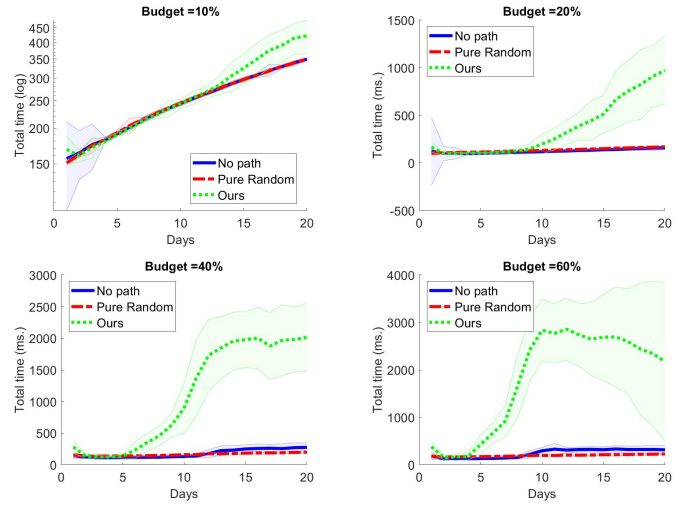


Fig. 4. Total run time for building disease maps after each day for 20 days.

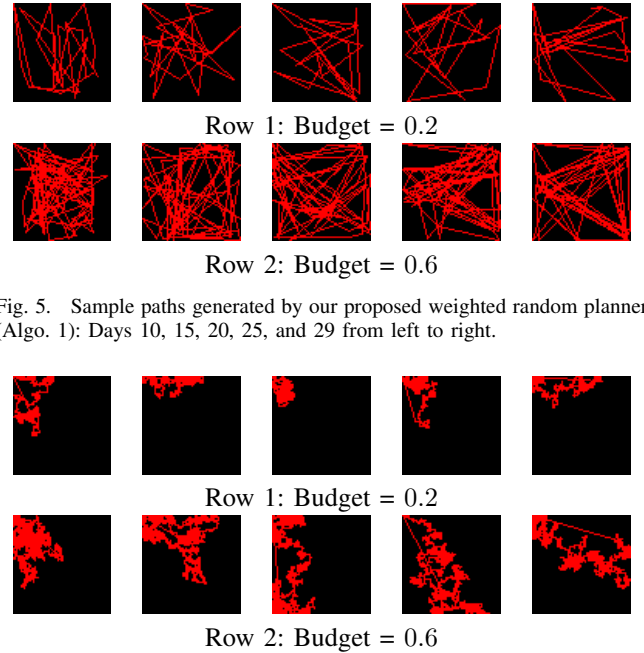


Fig. 5. Sample paths generated by our proposed weighted random planner (Algo. 1): Days 10, 15, 20, 25, and 29 from left to right.

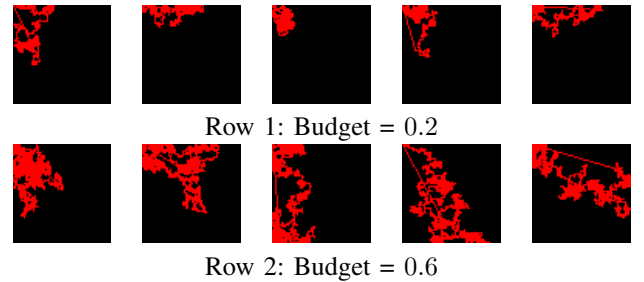


Fig. 6. Sample paths generated by the ‘Pure Random’ planner: Days 10, 15, 20, 25, and 29 from left to right.

proposed framework than the two tested baselines, namely ‘No Path’ and ‘Pure Random’ (Fig. 4). This is a trade-off as our proposed technique combines neural network-based prediction with an intelligent informed search strategy for path planning, it can model the disease state better, i.e., produces a lower-MSE disease map after  $n$  days while incurring higher costs for planning itself. Sample paths produced by our technique and the random baseline are shown in Figs. 5 and 6. As discussed previously, our proposed approach makes the robot explore the whole environment gracefully whereas the random baseline can be limited to one part of the field. Sample screenshots of disease spread maps created by the tested methods and the ground truth disease map are presented in Fig. 7.

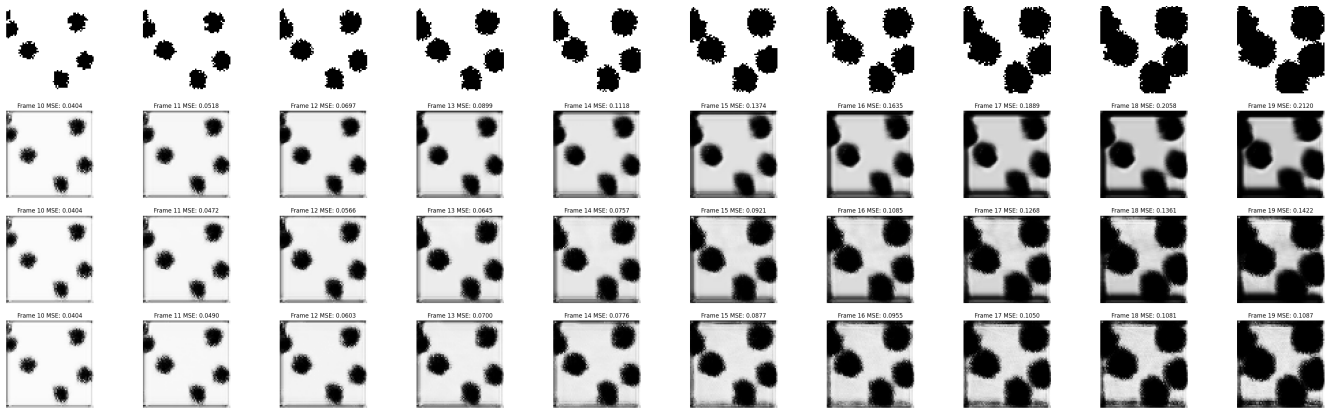


Fig. 7. Screenshots of disease maps for days 10 through 19 after the complete observation  $k = 10$  days. From top row to bottom: ground truth, ‘No Path’, ‘Pure Random’, and Ours.

## VI. CONCLUSION AND FUTURE WORK

Monitoring the spread of disease within agricultural environments is critically important, particularly in light of increasing population pressures and decreasing available farmland. This study introduced an innovative artificial intelligence framework designed to enhance food security through precise disease spread monitoring. We employed a combination of convolutional and recurrent neural network-based forecasting techniques alongside an informed search strategy that integrates the output of disease forecasts to plan intervention strategies. The proposed framework was tested in a simulated  $64 \times 64$  environment using the SIRI model to mimic disease propagation in crops. The results from our simulations demonstrate that our framework surpasses two existing baselines in both the accuracy of disease prediction and effectiveness in constructing disease maps. Looking ahead, we aim to explore the integration of a multi-robot system to assess potential improvements in prediction accuracy. Additionally, this paper operated under the assumption of available historical disease spread data for training our model. As a future endeavor, we plan to delve into the potential of unsupervised and reinforcement learning methods, which rely less on the availability of extensive training datasets. These efforts are anticipated to significantly advance the capabilities of autonomous agricultural disease management, catering to the demands of modern agricultural practices.

## REFERENCES

- [1] P. Jawade, D. Chaugule, D. Patil, and H. Shinde, “Disease prediction of mango crop using machine learning and iot,” in *Advances in Decision Sciences, Image Processing, Security and Computer Vision: International Conference on Emerging Trends in Engineering (ICETE)*, Vol. 1, pp. 254–260, Springer, 2020.
- [2] E. Moriones and J. Navas-Castillo, “Tomato yellow leaf curl virus, an emerging virus complex causing epidemics worldwide,” *Virus research*, vol. 71, no. 1–2, pp. 123–134, 2000.
- [3] D. F. Nettleton, D. Katsantonis, A. Kalaitzidis, N. Sarafjanovic-Djukic, P. Puigdollers, and R. Confalonieri, “Predicting rice blast disease: machine learning versus process-based models,” *BMC bioinformatics*, vol. 20, pp. 1–16, 2019.
- [4] S. S. Shinde and M. Kulkarni, “Review paper on prediction of crop disease using iot and machine learning,” in *2017 International conference on transforming engineering education (ICTEE)*, pp. 1–4, IEEE, 2017.
- [5] A. Dutta, S. Roy, O. P. Kreidl, and L. Bölöni, “Multi-robot information gathering for precision agriculture: Current state, scope, and challenges,” *Ieee Access*, vol. 9, pp. 161416–161430, 2021.
- [6] J. Su, C. Liu, X. Hu, X. Xu, L. Guo, and W.-H. Chen, “Spatio-temporal monitoring of wheat yellow rust using uav multispectral imagery,” *Computers and electronics in agriculture*, vol. 167, p. 105035, 2019.
- [7] S. Matloob, P. P. Datta, O. P. Kreidl, A. Dutta, S. Roy, and L. Bölöni, “Waterberry farms: A novel benchmark for informative path planning,” *arXiv preprint arXiv:2305.06243*, 2023.
- [8] M. A. Martínez-Beneito, A. López-Quilez, and P. Botella-Rocamora, “An autoregressive approach to spatio-temporal disease mapping,” *Statistics in medicine*, vol. 27, no. 15, pp. 2874–2889, 2008.
- [9] W. Xu, Q. Wang, and R. Chen, “Spatio-temporal prediction of crop disease severity for agricultural emergency management based on recurrent neural networks,” *GeoInformatica*, vol. 22, pp. 363–381, 2018.
- [10] P. Skelsey, “Forecasting risk of crop disease with anomaly detection algorithms,” *Phytopathology*, vol. 111, no. 2, pp. 321–332, 2021.
- [11] J. Franke and G. Menz, “Multi-temporal wheat disease detection by multi-spectral remote sensing,” *Precision Agriculture*, vol. 8, pp. 161–172, 2007.
- [12] P. Ghosh, D. Mandal, A. Bhattacharya, M. K. Nanda, and S. Bera, “Assessing crop monitoring potential of sentinel-2 in a spatio-temporal scale,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 227–231, 2018.
- [13] L. E. de Oliveira Aparecido, G. de Souza Rolim, J. R. da Silva Cabral De Moraes, C. T. S. Costa, and P. S. de Souza, “Machine learning algorithms for forecasting the incidence of coffea arabica pests and diseases,” *International Journal of Biometeorology*, vol. 64, pp. 671–688, 2020.
- [14] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] S. Matloob, A. Dutta, P. Kreidl, D. Turgut, and L. Bölöni, “Exploring the tradeoffs between systematic and random exploration in mobile sensors,” in *Proceedings of the Int’l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pp. 209–216, 2023.