

Automated Wheat Disease Detection Using A ROS-Based Autonomous Guided UAV

Behzad Safarjalal

K.N.Toosi University of Technology

Yousef Alborzi

University of Manitoba

Esmail Najafi (✉ najafi.e@kntu.ac.ir)

K.N.Toosi University of Technology

Research Article

Keywords:

Posted Date: March 7th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1251771/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Automated Wheat Disease Detection using a ROS-based Autonomous Guided UAV

Behzad Safarijalal^{1,+}, Yousef Alborzi^{1,2,+}, and Esmail Najafi^{1,*}

¹K.N.Toosi University of Technology, Mechanical Engineering , Tehran, Iran

²University of Manitoba, Mechanical Engineering, Winnipeg, Canada

*najafi.e@kntu.ac.ir

⁺these authors contributed equally to this work

ABSTRACT

With the increase in world population, food resources have to be modified to be more productive, resistive, and reliable. Wheat is one of the most important food resources in the world, mainly because of the variety of wheat-based products. Wheat crops are threatened by three main types of diseases which cause large amounts of annual damage in crop yield. These diseases can be eliminated by using pesticides at the right time. While the task of manually spraying pesticides is burdensome and expensive, agricultural robotics can aid farmers by increasing the speed and decreasing the amount of chemicals. In this work, a smart autonomous system has been implemented on an unmanned aerial vehicle to automate the task of monitoring wheat fields. First, an image-based deep learning approach is used to detect and classify disease-infected wheat plants. To find the most optimal method, different approaches have been studied. Because of the lack of a public wheat-disease dataset, a custom dataset has been created and labeled. Second, an efficient mapping and navigation system is presented using a simulation in the robot operating system and Gazebo environments. A 2D simultaneous localization and mapping algorithm is used for mapping the workspace autonomously with the help of a frontier-based exploration method.

Introduction

As the world's population increases, the demand for nutrition resources increases, while in many places, there are no productive farmlands left to use. As a result, the need for ways to produce more on established farmland and more products from fewer resources is one of the present concerns of mankind. One solution is to utilize more efficient technologies in agriculture and use robots instead of human labor in specific tasks to increase production. These include crop scouting, pest and weed control, harvesting, targeted spraying, and sorting. These robots can be classified into two main categories in general: unmanned ground vehicles and unmanned aerial vehicles.

Like many other new technologies, the first use of a unmanned aerial vehicle (UAV) was for military purposes¹. However, today the applications of these aerial robots can be found in many civilian sectors, such as agriculture. They are used in almost any kind of agricultural tasks such as crop monitoring and disease detection^{2,3}, crop height estimation⁴, pesticide spraying⁵ and soil and field analysis⁶. Although recent developments in robotic research have made these machines very smart and effective, the need for further improvements to make them more efficient and more intelligent is obvious and unavoidable.

Regardless of the plant's type, pests have always been one of the biggest concerns of farmers. The attempts for increasing final products by eliminating destructive pests or weeds are age-old concerns. Recent developments in chemical pesticides (mainly after world war II), have brought food safety to humans by removing these irritating creatures and decreasing food loss. On the other hand, the overuse of chemical substances and especially pesticides, has raised concerns about their consequences on human health, which have caused global efforts for chemical usage reduction in food industries. One of the major outcomes of these efforts is a process called integrated pest management (IPM). IPM is concerned with controlling and reducing the number of pests to an acceptable extent, by performing effective, economical, and feasible methods and actions. Some of these actions include using resistant varieties, and natural predators and parasites instead of chemical substances⁷. However, sometimes the employment of pesticides is inevitable. In these cases, using a precise and smart system, which can minimize the amount of used chemical pesticides, by on-time plant disease detection and accurate pesticide spraying on the infected plants, instead of spraying the whole farmland, can help produce more healthy products.

Rusts are one of the most common wheat diseases in the world. In general, three types of rust diseases can damage wheat crops: stem rust, stripe rust, and leaf rust. Wheat leaf rust, caused by *Puccinia Triticina*, leads to a loss in product yield and quality. Also known as brown rust, this disease is usually less harmful than the other types of rust. However, because of its widespread existence, it amounts to more annual damage overall⁸. In some regions, the amount of damage caused by this type of disease can reach values of up to 70%⁹. Stripe rust, also known as yellow rust, is caused by *Puccinia Striiformis*. Regional

yield losses due to stripe rust, have been reported to reach amounts of 25% while this amount rose to numbers as high as 80% during the Middle East and North African epidemic in 2010¹⁰. While less common, stem rust or black rust, caused by *Puccinia Graminis*, can result in serious losses up to 33% of the final products⁹. The amount of inflicted damage by these rusts mainly depends on species resistance and climate situations. The purpose of this work is to propose a smart UAV system, which can monitor a wheat farmland autonomously. This system consists of two parts: an autonomous control and navigation system, and a wheat rust detection system, to monitor the wheat farmland and detect the corrupted crops respectively. The control and navigation system is implemented and tested in a simulated environment using the robot operating system (ROS) and Gazebo. A preliminary version of this work has been already published in¹¹. The paper is organized as follows. First, An image-based wheat disease detection and classification system is presented, followed by a description of the implemented quadrotor system structure and algorithms. Finally, the performance of the quadrotor system and the results of the proposed deep learning image classification methods are presented and discussed.

Methods

The on-time detection of plant diseases is of vital importance in decreasing losses in crop yield. Although laboratory testing is the most accurate way of detecting diseases in plants, it is not always available to farmers and costs a substantial amount. However, because most plant diseases are associated with some visible feature, studying these visual patterns on the plants is a less expensive way of detecting the infections.

Plant disease detection

Recent advances in image processing and deep learning algorithms have inspired their popularity and drove many teams and researchers to use these techniques for various applications. Additionally, the rise of deep learning algorithms has caused improvements in the accuracy and speed of many problems, especially in image classification. This has led to a great number of researches for Automatic plant disease detection systems based on image classifiers. While earlier works such as¹² use conventional image processing techniques to extract features such as color, important edges, and histograms to detect plant diseases, machine learning algorithms utilize available data to implement more robust models. For example, in¹³, K-means clustering along with feature extraction and an SVM classifier have been used to detect diseases in grape images with an 88.89% accuracy.

More recently, with the help of newly developed Graphical Processing Units (GPU), deep learning techniques, especially CNNs, have experienced great success in image classification problems¹⁴ and have become the state of the art in computer vision. Their success on plant disease detection datasets is shown in several previous works. In¹⁵, five different CNN models have been trained and tested on a large dataset, called Plant Village, with 58 classes of plants and diseases. The authors of¹⁶ used variations of the MobileNet¹⁷ architecture to achieve excellent results on the Plant Village dataset. In¹⁸ the VGG-16 architecture has been used to identify five major diseases in eggplant leaves based on a novel dataset. The authors also analyzed the effects of using images in different colors spaces on the accuracy of the system. A different CNN-based approach has been implemented on a custom dataset of different plants in¹⁹, in which individual lesions and spots on the leaf are considered for disease detection, rather than using the whole leaf as input. Furthermore, hyperspectral imaging has also proven to be effective for plant disease detection in different scales in papers such as²⁰ and²¹. This work proposes a two-stage classifier to enable our UAV to detect plant diseases. First, the model uses an object detection network to find individual plant leaves in the image, and then after cropping the image with the bounding box coordinates, the second classifier detects the type of the disease on the leaf.

Dataset description

To the best of our knowledge, there is no open-source dataset of wheat plants with the mentioned diseases available. Therefore, to train our models, we created a custom dataset. The dataset contains 900 images of healthy and infected wheat plant leaves captured in real cultivation conditions in commercial farms. The images were first annotated by hand by bounding boxes for the object detection algorithms. In each image we have annotated as many leaves as possible by specifying their region and labeling all of them under one class regardless of their visual health condition. And for the second-stage classifier, 3672 images of individual leaves were cropped by hand and labelled with their corresponding disease for each class. Finally, they were split into train, validation and test sets randomly. A sample from every class is shown in Fig. 1 and the dataset is described in more details in Table 1.

Wheat leaf detection

To detect individual leaves in the image, we require an object detection network. Currently, the state of the art in object detection networks are the YOLO V4²² and EfficientDet²³ models. The YOLO V4 model consists of a CSPdarknet53²⁴ backbone and the YOLO V3²⁵ head for the detector while also implementing different Bag of Freebies and Bag of Special methods such as Mosaic data augmentation, DropBlock²⁶ and CutMix²⁷ regularization. In the Efficientdet paper, by utilizing the Efficientnet architecture²⁸ as the backbone and a bi-directional feature pyramid network (BIFPN) as the feature network and prediction

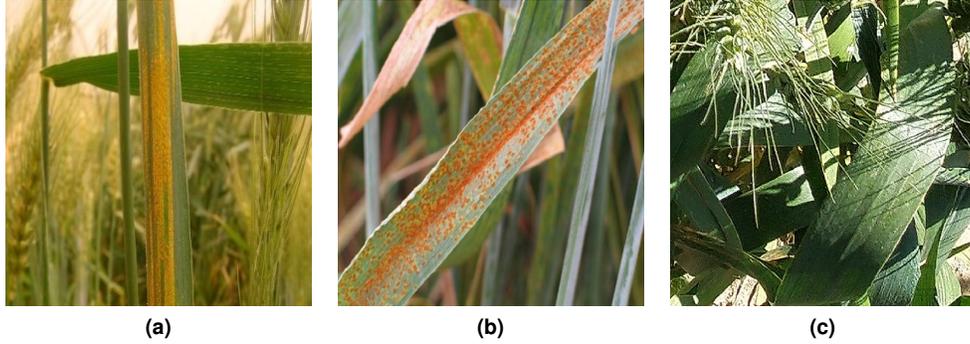


Figure 1. Samples from the collected wheat disease dataset: (a) yellow-rust-infected wheat leaf, (b) brown-rust-infected wheat leaf, (c) healthy wheat leaf

Set	Brown Rust	Yellow Rust	Healthy
Train	894	928	1116
Valid	111	116	139
Test	112	116	140
Total	1117	1160	1395

Table 1. The details of each class and set of our wheat disease dataset

networks, the authors were able to achieve similar accuracies as state-of-the-art models while being 4x-9x smaller. Here we have implemented 4 different object detection networks and trained them on our annotated dataset. Because the features of our plant diseases are relatively small, the network requires images with somewhat high resolutions, therefore, we have trained the networks with 800×800 resolution images. the training parameters are kept constant for all of the models so that the results can be compared fairly.

Wheat rust classification

After the leaf bounding boxes are detected on the UAV acquired image, they are cropped and fed to the classification network. For the classification part of the detection system, we tested multiple approaches to get the best results while keeping in mind the computation cost. With their weight-sharing ability, CNNs are the best candidate for image classification. In this work, we implemented five different CNNs which were MobileNet V3, VGG 16, Inception, ResNet 50, and EfficientNet-B0. The models are trained for 50 epochs using the adam optimizer with an initial learning rate of 0.001 and on each epoch, the learning rate is divided by a factor equal to the epoch count. Furthermore, data augmentation techniques were used to increase the number of training data by randomly applying rotation, zoom, brightness change, and shearing.

UAV navigation and control

Autonomous navigation of a robot in an arbitrary location requires various tools and consists of different processes. These components can vary in different cases based on many parameters such as the type of the robot, its workspace (indoor or outdoor), and utilized sensors. Probably the first requirement is a map, which contains the static obstacles of the robot's workspace. This map can be a 2D or 3D representation of the robot's working environment and is acquired in a process called mapping. It is also possible to obtain this map using the robot itself with the help of another process called localization, which enables the robot to map its environment while simultaneously estimating its own position regarding this map. This estimation gets more accurate as the robot moves around and gathers more information about its surrounding. The tasks of mapping and Localization are performed using sensors such as a camera, a laser range-finder, or an ultrasonic sensor, which can help the robot evaluate its distance to the nearby obstacles. By executing mapping and localization together at the same time, the robot is able to map its working environment. This simultaneous performance is referred to simultaneous localization and mapping (SLAM)²⁹. Furthermore, this job can be done by manually moving the robot through its workspace until the acquired map is completed, or it can also be done autonomously. Autonomous SLAM is possible with the help of Exploration, in which the robot explores its environment automatically, to map the whole area.

All of the mentioned techniques can be implemented and tested in a simulation environment. The robot operating system ROS with the help of Gazebo and Rviz can be utilized to do such a simulation. ROS is an open-source operating system with several styles of communication and great package management. In addition to lots of packages for almost any type of robotic

applications, ROS supports many programming languages such as C++ and python. The main purpose of ROS's development is to support code reuse in robotics research, so anyone could be able to use and improve its materials. This has caused it to become one of the most popular robots simulating environments and its community is growing every day. Furthermore, Gazebo is an excellent free tool that can be used to create various indoor and outdoor environments (worlds) and visually simulate robot movement and navigation in these environments.

Localization and mapping

The employed drone for this work must be able to carry the payload of an appropriate RGB camera and a laser range-finder to perform the detection and navigation tasks respectively. In this case, a simple and suitable choice would be Parrot A.R. Drone 2.0. There are many packages available regarding robot simulation for ROS. The tum-simulator package provides a good simulated model of Parrot A.R. Drone 2.0. In this study, this model is used along with hector-quadrator package, to simulate the drone and implement the requisite systems for autonomous navigation. Hector-quadrator, provided by Team Hector, is one of the most popular and powerful packages regarding the simulation and navigation of drones³⁰. Moreover, reducing the computation costs of the navigation and control process as much as possible is a crucial task. To achieve this goal, using 2D SLAM and navigation methods instead of 3D ones, could be a more efficient and sufficiently accurate approach. So, the "gmapping" and "amcl" packages were utilized to implement OpenSlam's Gmapping and Monte-Carlo localization³¹ algorithms in ROS respectively. The "gmapping" package can construct an occupancy grid map efficiently using data provided by the mounted laser rangefinder sensor, based on Rao-Blackwellized particle filters³². Also, the "amcl" package can perform the localization task by the adaptive (or KLD-sampling) Monte-Carlo localization method.

Navigation

There are two path planners required for the navigation phase, namely global and local planners. The adapted move-base package uses the A* algorithm³³ to compute efficient paths toward goal points, which can handle static obstacles specified in the constructed map. Moreover, the "dwa-local-planner" package handles the dynamic obstacles, which are not provided in the map, by computing appropriate velocity commands using the dynamic window approach³⁴. Finally, by using the "exploration-lite" package³⁵ along with the earlier mentioned packages, the mapping process can be done autonomously. This package's functionality is based on frontiers³⁶, which are specific points in the uncompleted occupancy grid maps that are between unoccupied and unmapped cells. The robot tries to move toward these frontiers to map the unknown environments, until there are no more frontiers, and thus no more unmapped areas left in its workspace. As mentioned before, the utilized packages for the robot navigation in this work are all 2D. To be more specific, they are designed for UGV navigation, and thus unable to control the drone's height. To control the robot's altitude, a node was written to lift the drone by sending appropriate velocity commands and keep the drone's altitude fixed during its flight using a PID controller.

System architecture

In this ROS-based system, different processes are carried out within ROS nodes, which interact with each other using ROS topics. As presented in Fig. 2, the autonomous mapping procedure is done by the cooperation of the "slam-gmapping" and

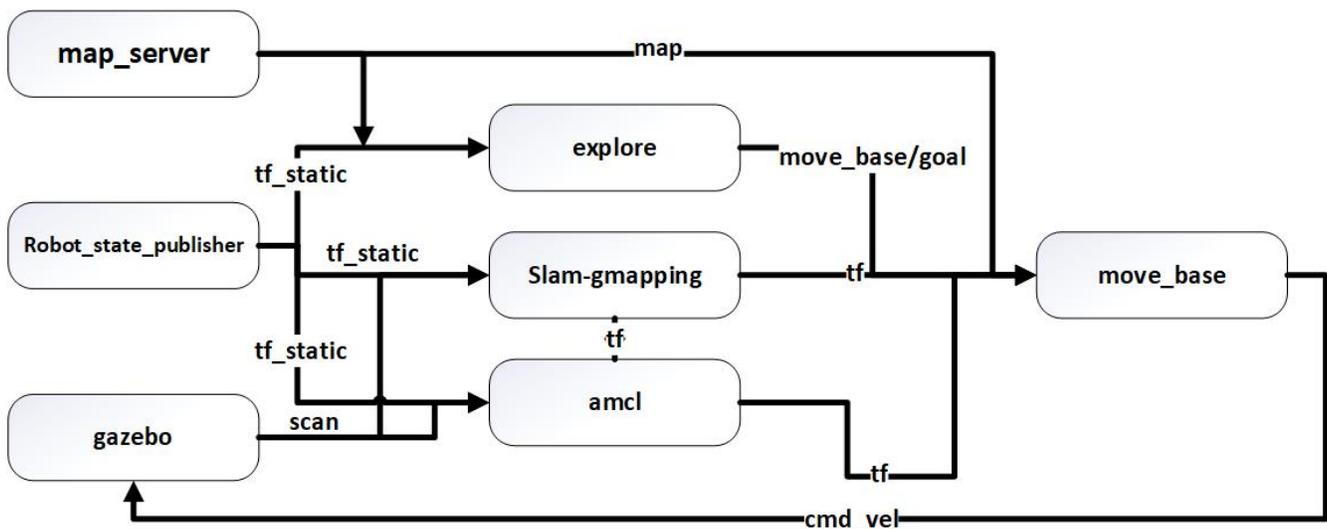


Figure 2. UAV navigation system architecture in ROS

Model	mAP	FPS	BFLOPs
YOLOV4	15	13.3	220.278
YOLOV4 tiny	19	42.3	25.101
YOLOV4 tiny-3 yolo layers	16.50	41.6	29.652
Efficientdet-B0	10.65	16.4	13.569

Table 2. Results of different object detection models on our wheat disease dataset

"explore" nodes. By acquiring the drone's current pose via the "tf" topic and the laser rangefinder measurements via the "scan" topic, the "slam-gmapping" node updates the partially-constructed map. On the other hand, by subscribing to the "tf" and "map" topics, the "explore" node, finds the frontiers and sends pose goals to the "move-base" node. The "move-base" node computes the appropriate velocity commands for the robot and creates the local and global costmaps. Finally, the velocity commands are sent to the gazebo node for visualization purposes.

Results

With the integration of the disease detection system and efficient UAV navigation and mapping, farmers can autonomously monitor their lands for diseases. In this process first, a 2D map of the land is acquired which contains the location and type of the infected crops. This information can then be used to perform appropriate actions, such as spraying a pesticide. Although we designed and trained our detection networks on a dataset for wheat diseases, the proposed system can easily be adapted to any other kind of plant by simply using the appropriate dataset. The results of the proposed wheat disease detection and UAV navigation systems are presented in this section.

Image-based disease detection

In Table 2, we report the metrics of the different implemented object detection networks. Taking into consideration the importance of speed for our system, we conclude that the YOLOV4 tiny model is the best fit. It is clear that the YOLO V4-tiny model achieves the best results with the highest mAP and FPS. An instance of this model on a sample of our dataset is displayed in Fig. 3.

Next, Table 3 compares the different CNN models in terms of performance and their accuracy on the disease classification test set. Looking at Table 3, we can see that, while the MobileNet-V3 model requires fewer floating point operations for running an instance, the EfficientNet-B0 model has the best combination of accuracy and GFLOPs among all of the networks. For this model, we present the evaluation matrix containing the precision, recall and F1-score, in Table 4 and the confusion matrix in Fig. 4. The confusion matrix shows that the EfficientNet-B0 model only had a single incorrect prediction in the whole test set.



Figure 3. Instance of leaf detection from YOLOV4-tiny model

Model	Accuracy	GFLOPs
MobileNet V3	94.29	0.174
Inception	98.64	5.69
ResNet-50	98.91	7.75
EfficientNet-B0	99.72	0.794
VGG16	99.45	31

Table 3. Results of different CNN classification models on our wheat disease dataset

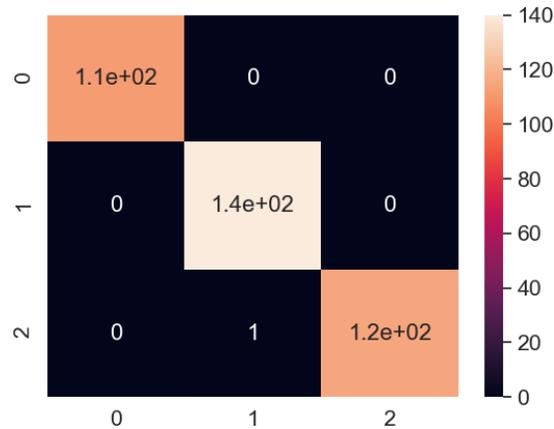


Figure 4. Confusion matrix for EfficientNet-B0 model on our wheat disease dataset

	Precision	Recall	F1-Score
Brown-rust	1.00	0.99	1.00
Yellow-rust	0.99	1.00	1.00
Healthy	1.00	1.00	1.00
Accuracy			1.00

Table 4. EfficientNet-B0 Matrix of Evaluation

Quadrotor mapping and navigation

To evaluate the mapping and navigation system, we carried out tests in the "willow-garage" gazebo simulated world. A visualization of the localization process using the "amcl" package is shown in Fig. 5 and as evident by the particle cloud condensation displayed in Fig. 5, after some movement by the robot, the uncertainty of the robot's pose decreases and the localization estimation becomes more accurate. Moreover, with the combination of the generated frontiers by the "exploration-lite" package, an accurate localization, and range measurements from the laser sensor, the SLAM algorithm acquired a map of the simulated environment as shown in Fig. 6. While the simulation environment appears much more complex than a wheat farmland, the mapping system was able to generate the map autonomously without any interference using the exploration package.

Using the constructed map and the implemented localization system, the UAV is able to readily navigate between two arbitrary points. Fig. 7 shows a successful instance of the A* path planning algorithm for an arbitrary goal pose in the simulated environment. Permission to reuse figures 5, 6 and 7 from the preliminary version of our work in¹¹ has been obtained from the publisher under the IEEE license number 5246641444156.

Discussion

For the object detection part of the system, by looking at Table 2, we can see that the mAP values of the trained networks seem low compared to the results of these networks on other datasets. This is mostly due to the fact that the desired objects in our dataset are close in color and shape to their background and it generally is a difficult task to detect the individual leaves in the scene. Also, in each image in the dataset there are a large number of wheat leaves present and there is the possibility of

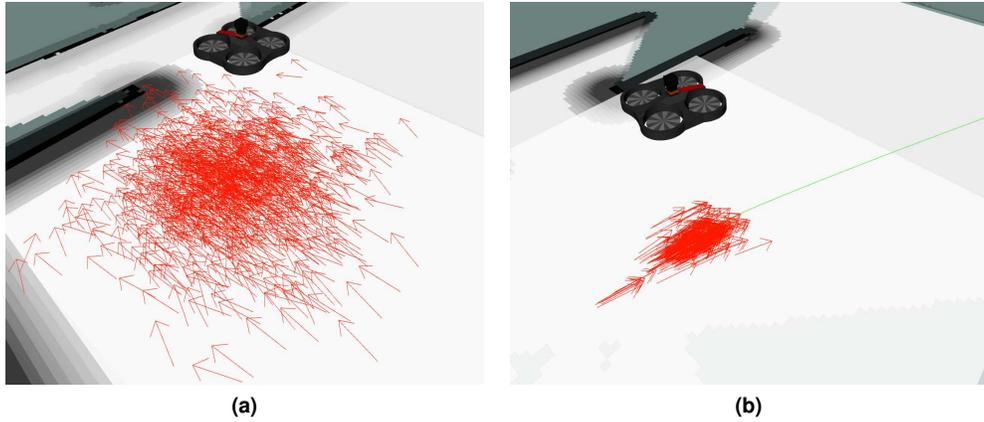


Figure 5. Visualized particle clouds condensation process of the "amcl" package (a) initial state (b) state after movement

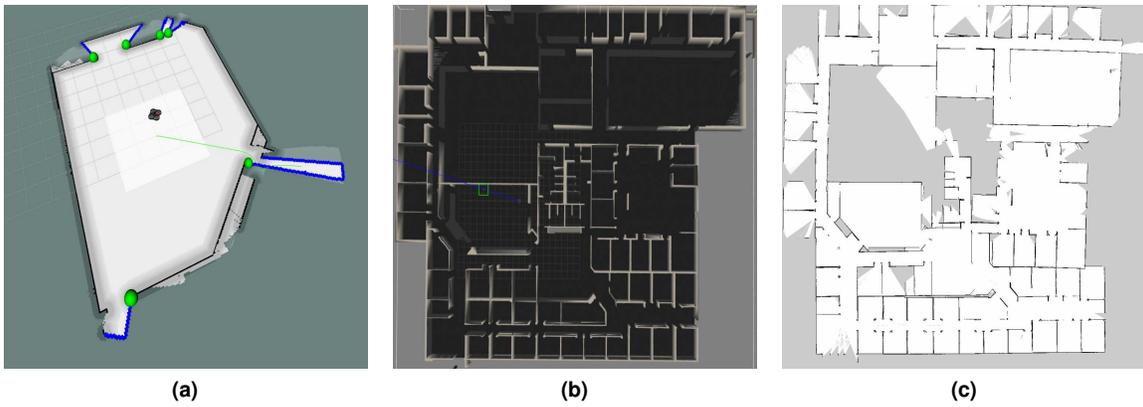


Figure 6. Exploration and mapping process : (a) Frontier points detected by the implemented algorithm (green points), (b) Simulated environment in Gazebo, (c) constructed map with frontier exploration

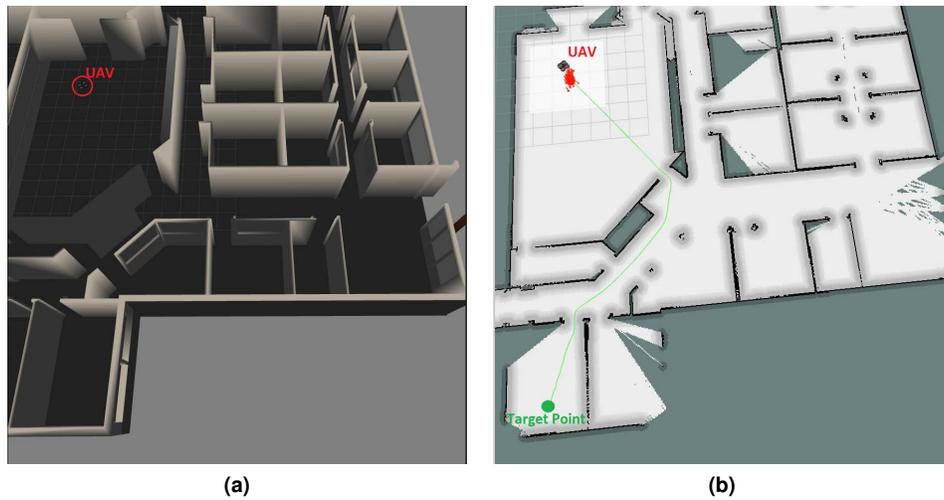


Figure 7. Simulated UAV navigation result: (a) UAV in simulated Gazebo world (b) arbitrary navigation path (green contour) in the same simulated environment

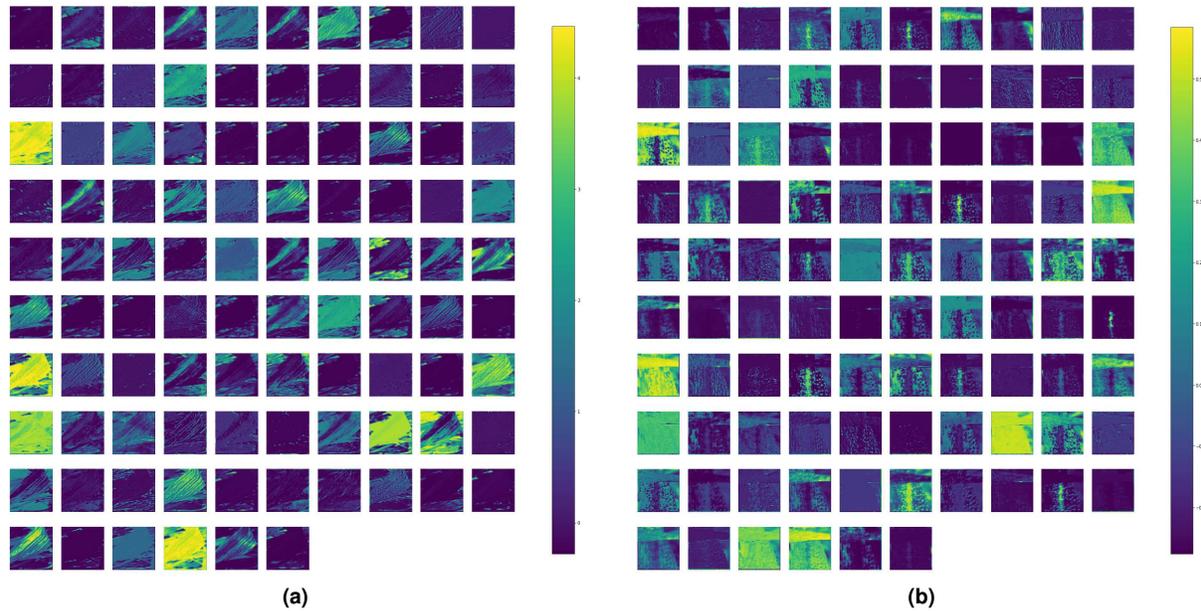


Figure 8. Feature map of the EfficientNet-B0 CNN model of (a) yellow-rust-infected leaf , (b) brown-rust-infected leaf

missing a few leaves in the labeling process which would hurt the mAP score of the object detection networks. Nevertheless, for our application we do not need to detect every single leaf in the image captured by the UAV. Therefore the networks seem to perform sufficiently in detecting enough leaf samples in an image, as shown in Fig. 3.

Table 3 shows that the classification networks achieve high accuracies on the test set after the 50 epochs of training. Additionally, visualizing the intermediate feature maps of different blocks of the model can shed light to what the network is actually learning. In Fig. 8, the feature maps of the second convolutional block of the efficientNet-B0 model are displayed. These figures show that the network has been able to learn to detect dotted features for brown rust infected leaves and striped shape features in Yellow-rust infected leaves.

Conclusion

In this work, a smart wheat crop monitoring system has been proposed. The image-based classification system can detect individual leaves in the images captured by the UAV mounted camera and classify them into three groups of brown-rust-infected, yellow-rust-infected, and healthy leaves. After evaluating different object detection algorithms on our custom dataset, the YOLOV4-tiny model outperformed other algorithms and achieved the best results with a 19% mAP 0.5 and 42 frames per second frame rate. For the disease classification system, the performance of the EfficientNet-B0 model was superior to other tested CNN architectures, recording the highest accuracy of 99.72% and 0.794 GigaFLOPs. Finally, an efficient navigation system based on the A* algorithm was implemented on the UAV, which can find safe trajectories between two arbitrary points in the map. The required map of the environment is also acquired autonomously by using the SLAM algorithm and the frontier exploration technique. This navigation system was simulated in Gazebo using the robot operating system.

Data Availability

The dataset generated and analysed during the current study are available in the Kaggle repository via the following web link: <https://www.kaggle.com/sinadunk23/behzad-safari-jalal>

References

1. Fahlstrom, P. & Gleason, T. *Introduction to UAV systems* (John Wiley & Sons, 2012).
2. Sugiura, R., Noguchi, N. & Ishii, K. Remote-sensing technology for vegetation monitoring using an unmanned helicopter. *Biosyst. engineering* **90**, 369–379 (2005).
3. Görlich, F. *et al.* Uav-based classification of cercospora leaf spot using rgb images. *Drones* **5**, 34 (2021).

4. Anthony, D., Elbaum, S., Lorenz, A. & Detweiler, C. On crop height estimation with uavs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4805–4812 (2014).
5. Huang, Y., Hoffmann, W. C., Lan, Y., Wu, W. & Fritz, B. K. Development of a spray system for an unmanned aerial vehicle platform. *Appl. Eng. Agric.* **25**, 803–809 (2009).
6. Primicerio, J. *et al.* A flexible unmanned aerial vehicle for precision agriculture. *Precis. Agric.* **13**, 517–523 (2012).
7. Damalas, C. A. Understanding benefits and risks of pesticide use. *Sci. Res. Essays* **4**, 945–949 (2009).
8. Roelfs, A. P. & Bushnell, W. R. *The cereal rusts*, vol. 2 (Academic Press Orlando, FL, 1985).
9. Singroha, G., Reddy, G., Gupta, V. & Kumar, S. *Wheat Diseases and Their Management*, 348–372 (2017).
10. Solh, M., Nazari, K., Tadesse, W. & Wellings, C. The growing threat of stripe rust worldwide. In *Proceedings of the Borlaug Global Rust Initiative Conference*, 1–4 (2012).
11. Alborzi, Y., Safari, B. & Najafi, E. ROS-based SLAM and navigation for a Gazebo-simulated autonomous quadrotor. In *Proceedings of the 21st International Conference on Research and Education in Mechatronics*, 1–5 (2020).
12. Bankar, S., Dube, A., Kadam, P. & Deokule, S. Plant disease detection techniques using canny edge detection & color histogram in image processing. *Int. J. Comput. Sci. Inf. Technol* **5**, 1165–1168 (2014).
13. Padol, P. B. & Yadav, A. A. Svm classifier based grape leaf disease detection. In *Proceedings of the Conference on advances in signal processing*, 175–179 (2016).
14. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Adv. neural information processing systems* **25**, 1097–1105 (2012).
15. Ferentinos, K. P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **145**, 311–318 (2018).
16. Kamal, K., Yin, Z., Wu, M. & Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **165**, 104948 (2019).
17. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520 (2018).
18. Rangarajan, A. K. & Purushothaman, R. Disease classification in eggplant using pre-trained vgg16 and msvm. *Sci. reports* **10**, 1–11 (2020).
19. Barbedo, J. G. A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **180**, 96–107 (2019).
20. Thomas, S. *et al.* Benefits of hyperspectral imaging for plant disease detection and plant protection: a technical perspective. *J. Plant Dis. Prot.* **125**, 5–20 (2018).
21. Golhani, K., Balasundram, S. K., Vadamalai, G. & Pradhan, B. A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* **5**, 354–371 (2018).
22. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection (2020). [2004.10934](https://arxiv.org/abs/2004.10934).
23. Tan, M., Pang, R. & Le, Q. V. Efficientdet: Scalable and efficient object detection (2020). [1911.09070](https://arxiv.org/abs/1911.09070).
24. Wang, C.-Y. *et al.* Cspnet: A new backbone that can enhance learning capability of cnn (2019). [1911.11929](https://arxiv.org/abs/1911.11929).
25. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement (2018). [1804.02767](https://arxiv.org/abs/1804.02767).
26. Ghiasi, G., Lin, T.-Y. & Le, Q. V. Dropblock: A regularization method for convolutional networks (2018). [1810.12890](https://arxiv.org/abs/1810.12890).
27. Yun, S. *et al.* Cutmix: Regularization strategy to train strong classifiers with localizable features (2019). [1905.04899](https://arxiv.org/abs/1905.04899).
28. Tan, M. & Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks (2020). [1905.11946](https://arxiv.org/abs/1905.11946).
29. Bailey, T. & Durrant-Whyte, H. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine* **13**, 108–117 (2006).
30. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U. & Von Stryk, O. Comprehensive simulation of quadrotor uavs using ROS and Gazebo. In *Proceedings of the International conference on simulation, modeling, and programming for autonomous robots*, 400–411 (2012).
31. Thrun, S., Burgard, W. & Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)* (The MIT Press, 2005).

32. Grisetti, G., Stachniss, C. & Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics* **23**, 34–46 (2007).
33. Hart, P. E., Nilsson, N. J. & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Syst. Sci. Cybern.* **4**, 100–107 (1968).
34. Fox, D., Burgard, W. & Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robotics & Autom. Mag.* **4**, 23–33 (1997).
35. Hörner, J. Map-merging for multi-robot system. (2016).
36. Yamauchi, B. A frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 146–151 (1997).

Author contributions

B.S. and E.N. collected dataset. Y.A. and B.S. implemented methods, analysed the results and drafted manuscript. All authors reviewed the manuscript.

Acknowledgments

The authors would like to thank the SharifAgRobot team for their support in collecting the dataset used in this work.

Competing interests

The authors declare no competing interests.

Figures

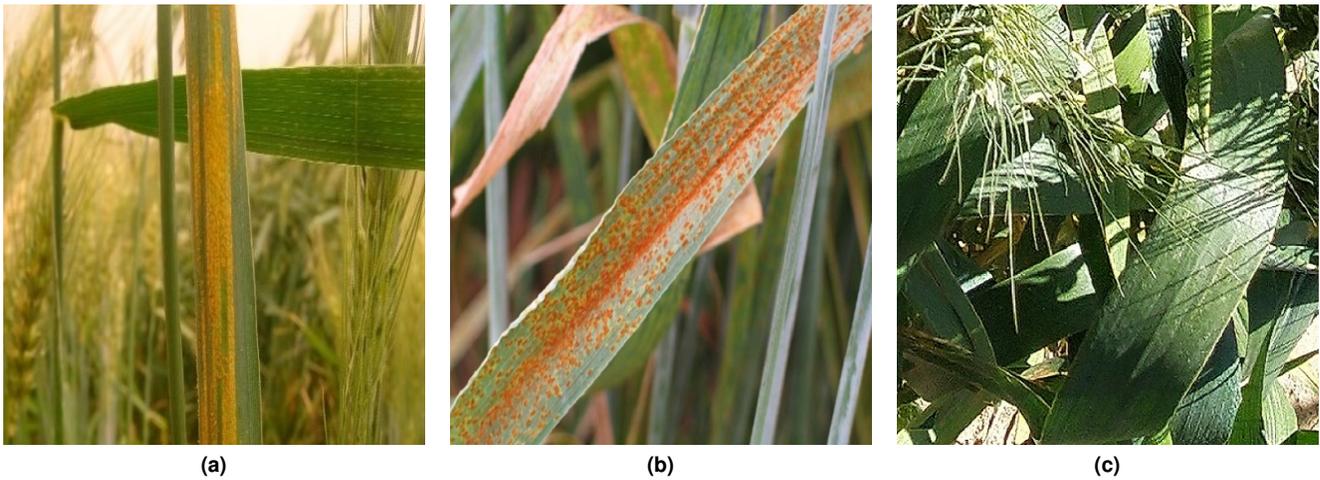


Figure 1. This figure shows a sample from each subset of our customly acquired dataset of wheat leaf diseases. Figure 1(a) shows a wheat leaf infected with Stripe rust (yellow rust), while Figure 1(b) shows a wheat leaf infected with Leaf rust (Brown rust). Finally Figure 1(c) displays a healthy wheat leaf with no visual signs of diseases.

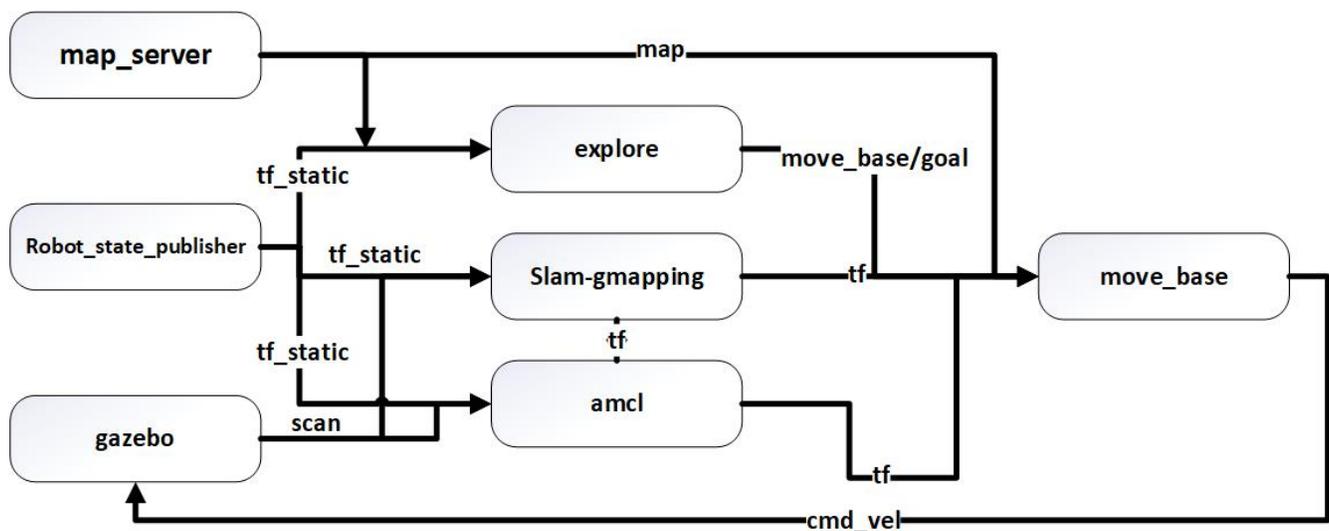


Figure 2. This figure illustrates the underlying communication layout of the simulated Parrot Ar.Drone 2.0 UAV in the Robot Operating System environment. ROS nodes, represented by box shapes, communicate with each other by sending and receiving messages through topics, represented by arrows.



Figure 3. This figure shows a sample output of the YOLOV4-tiny model on an instance of the leaf detection dataset. The purple bounding boxes show detected leaves in the image and the percentage values on the top right show the confidence score of the network for each instance. Although due to the high complexity of the object backgrounds, not every single leaf is detected by the deep learning model, the system is still able to detect enough samples in each image of the dataset in order to determine the health conditions of wheat plants in an area.

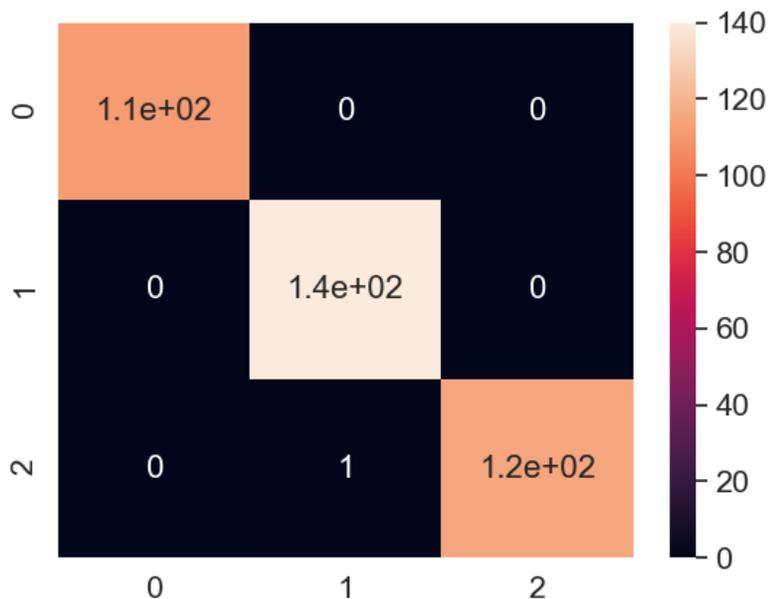


Figure 4. This figure shows the confusion matrix for EfficientNet-B0 model on our wheat disease dataset. The main diagonal elements of this matrix show the number of true predictions by the CNN model while off-diagonal elements represent false positive and false negative predictions. The matrix indices correspond to Brown rust infected, healthy and yellow-rust infected leaf image classes.

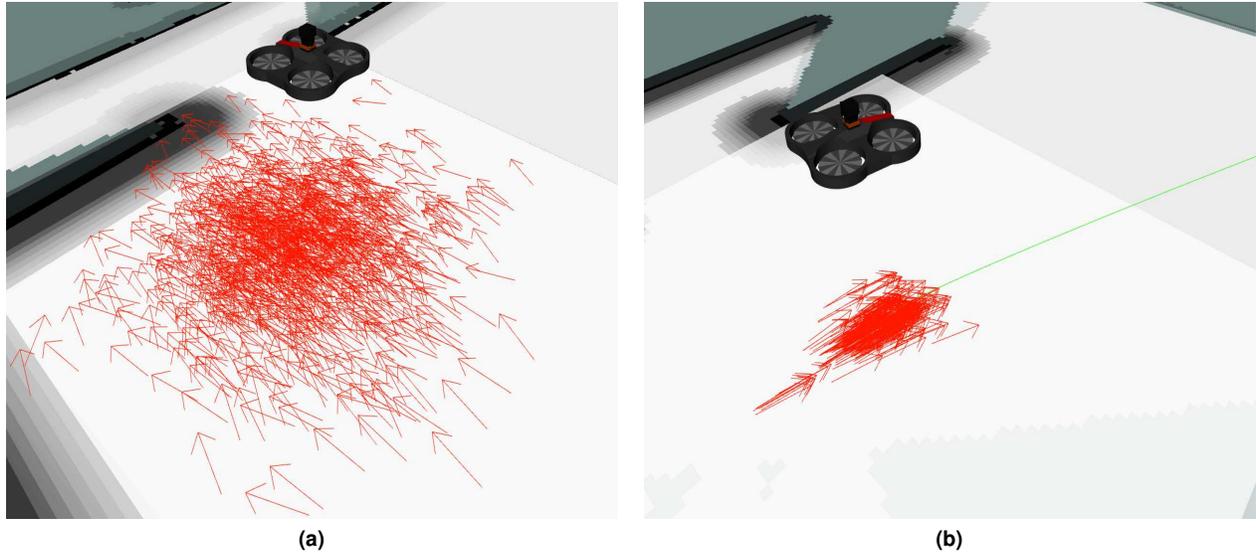


Figure 5. This figure shows the visualized particle clouds condensation process of the "amcl" package. Figure 5(a) shows the initial state of the localization estimate of the robot, where the randomly directed arrows are sparsely distributed, while Figure 5(b) shows a more condensed particle cloud of the same robot after some movement and gaining more certainty about the localization estimation¹¹

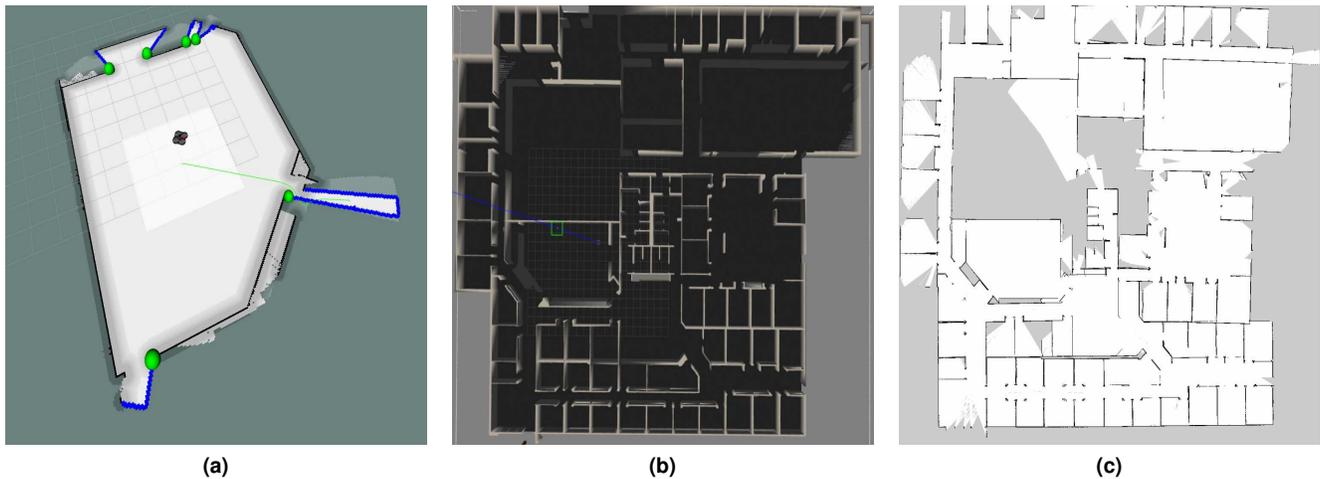


Figure 6. This figure displays the results of the exploration and mapping process. Figure 6(a) visualizes the frontier points, areas between unoccupied and unknown cells of the map, detected by the exploration-lite package. Also as shown in this figure, the robot moves towards the frontier point which best expands the map (shown by the green line). Figure 6(b) shows a top view of the simulation environment while Figure 6(c) shows the constructed map using the implemented SLAM algorithm.¹¹

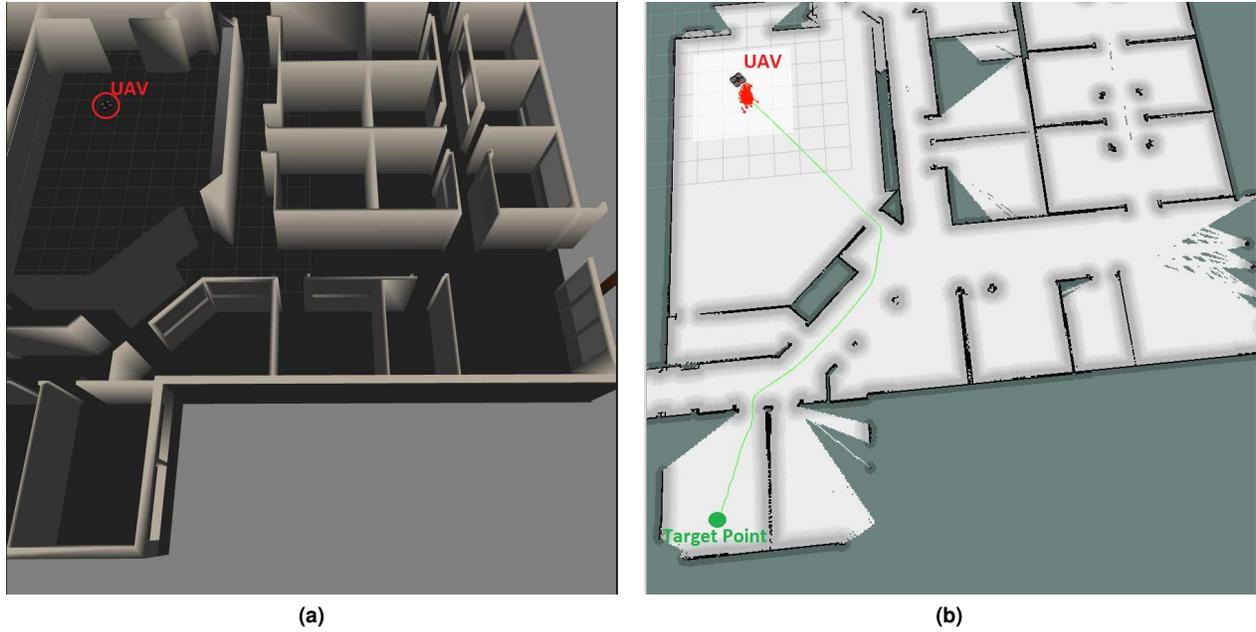


Figure 7. This figure shows an example of a generated path in the navigation process. Figure 7(a) shows the UAV in an arbitrary position in the simulated workspace and Figure 7(b) highlights a safe and efficient path generated by the A* search algorithm to move to a second arbitrary point.¹¹

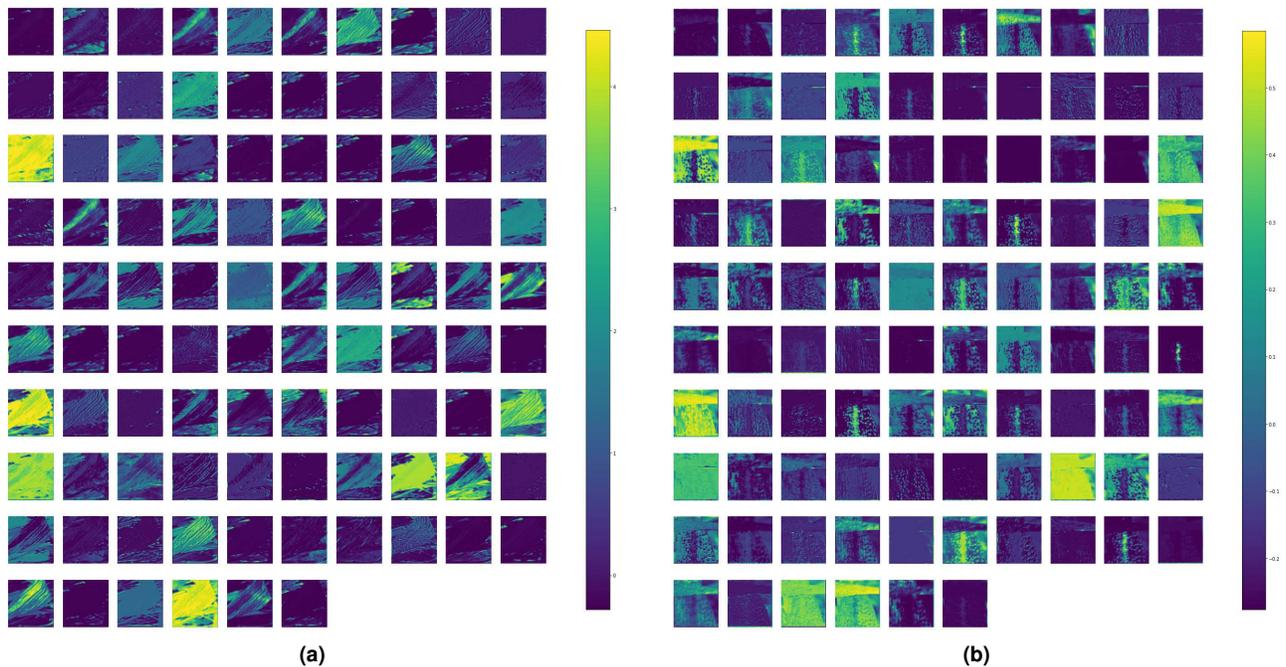


Figure 8. This figure shows a visualization of an intermediate layer of the EfficientNet-B0 CNN model. Figure 8(a) and 8(b) show the visualization of the said layer with a yellow-rust-infected leaf image and brown-rust-infected leaf image as input. Considering that the more yellow spots on the visualization show a higher activation of the network units, we can see that the network displays higher activations in some neurons when it sees stripe or dotted shape features, corresponding to yellow rust and brown rust visual patterns. This shows that the network has learned to recognize these visual patterns in the image.

Tables

Set	Brown Rust	Yellow Rust	Healthy
Train	894	928	1116
Valid	111	116	139
Test	112	116	140
Total	1117	1160	1395

Table 1. This table illustrates the details of the custom wheat disease dataset. The dataset consists of three classes of images, namely, Brown-rust-infected, Yellow-rust-infected and healthy wheat leaves. Images of each class are divided into three subsets for training, validation and testing with a 80%,10% and 10% split respectively.

Model	mAP	FPS	BFLOPs
YOLOV4	15	13.3	220.278
YOLOV4 tiny	19	42.3	25.101
YOLOV4 tiny-3 yolo layers	16.50	41.6	29.652
Efficientdet-B0	10.65	16.4	13.569

Table 2. This table shows information about different object detection models on our wheat leaf dataset in terms of accuracy and performance.

Model	Accuracy	GFLOPs
MobileNet V3	94.29	0.174
Inception	98.64	5.69
ResNet-50	98.91	7.75
EfficientNet-B0	99.72	0.794
VGG16	99.45	31

Table 3. This table compares the results of various convolutional deep learning classification models, trained and tested on our custom wheat disease dataset. The models are assessed in terms of accuracy and computational complexity (GFLOPs).

	Precision	Recall	F1-Score
Brown-rust	1.00	0.99	1.00
Yellow-rust	0.99	1.00	1.00
Healthy	1.00	1.00	1.00
Accuracy			1.00

Table 4. This table shows the precision, Recall and F-1-Score of the most efficient CNN model that we assessed on our custom wheat disease dataset. The EfficientNet-B0 model was able to achieve an almost perfect score in all of the mentioned performance metrics in all three classes. Note that the values reported in the table are rounded up to two decimal points.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [alborzi2020.pdf](#)