PAPER • OPEN ACCESS

Classification and Detection of Obstacles for Rover Navigation

To cite this article: Joshua H X Lim and Swee King Phang 2023 *J. Phys.: Conf. Ser.* **2523** 012030

Classification and Detection of Obstacles for Rover Navigation

Joshua H X Lim[1,2] and Swee King Phang[1,2,*]

[1] School of Engineering, Faculty of Innovation and Technology, Taylor's University, Subang Jaya, Selangor, Malaysia
[2] Digital Innovation & Smart Society Impact Lab, Taylor's University, Subang Jaya, Selangor, Malaysia

[*] corresponding author: SweeKing.Phang@taylors.edu.my

Abstract. In this research project, the author aims to achieve Level 3 conditional automation whereby the researched Unmanned Ground Vehicle (UGV) is bound to classify and detect its own obstacle with human assistance as it cruises through a plantation field. Recognizing the different classes of obstacles enable the UGV to plan out the most efficient path to meet its desired goal. The purpose of this research project was to develop a classification and detection of obstacle and an optimal path planning algorithm suitable to be implemented for relieving the working process in an extreme condition plantation field. This paper presents an algorithm whereby it can conduct image-based obstacle detection through image masking and model prediction, along with a trigonometrical-based path planning approach. The proposed algorithm should hypothetically allow the UGV to conduct real-time path planning as it classifies some common obstacles such as leaves, rocks, and branches existing in a plantation field. As the waypoints were marked from the Ground Control Station (GCS), the UGV will travel towards the given waypoints to complete the given mission. When the UGV meets an obstacle, it will first differentiate whether it's traversable, followed by running the proposed algorithm to avoid the risk of destructing the UGV by choosing a collision free path. The basic idea is to apply path planning by considering the available spacing between the detected obstacle by comparing with a predefined threshold. Through the provided threshold value, the UGV can identify the type of obstacle yet to be detected. For instance, obstacles within the given range of value can be labelled to be a leafy obstacle, otherwise it is not considered to be a leafy obstacle. To ensure the behaviour and safety measure of the UGV to run smoothly, the author had undergone model training for an elevated model prediction by training and deploying a custom training loop through TensorFlow. Nevertheless, MATLAB was utilized to test out the concept of the path planning algorithm to examine its behaviour as untraversable obstacles were met. All these implementations can further grow in the agricultural industries as it can aid humans with performing tedious and impossible tasks on site.

1. Introduction

In the Fourth Industrial Revolution (IR 4.0), humanity has gradually progressed into the agents of automation and Artificial Intelligence (AI). The need to achieve outputs in an efficient manner had rapidly change technology to develop a smarter system. Human beings need robots for dangerous, repetitive, and high-precision work; to save mankind from human imperfections and the impossible [1]. This leads to the application of UGV, an aberrated term that is overshadowed by the term "rover". Generally, a UGV has the capability to operate outdoors and over a wide variety of terrain. Moreover, functioning in the place of a human. Besides that, it can be deployed to conduct accurate ground target localization and travel through narrow areas.

Agriculture is quickly becoming a place for the development of high-tech robotic works. Attracting companies and professionals to investigate both the production capabilities along with the advancement in robotic and automation technologies. Furthermore, the utilization of automation and smart systems also aid in the safety of human's nourishment with the appropriate selection of chemicals on crops, the productivity to get task done in a fast pace while maintaining its quality, and even conducting repetitive task in extreme temperatures [2]. By means, providing the UGV proper anatomy and consciousness to hold the ability to act on its own. Therefore, to run a UGV autonomously, the choice of obstacle detection and path planning algorithm plays a crucial role.

Mobile robots are gradually expanding in number and complexity due to their all-rounder applications. They need locomotion mechanisms that allow them to traverse around their respective environments. Navigation in a dynamic environment was said by Vijaya and Nagaraja to be a complicated issue as a group of various kinds of obstacles were yet to be chosen to avoid any possible collision [3]. Based on their studies, an image-based obstacle detection and path planning system were proposed to estimate the traversable path for the autonomous robot. With the use of a microcontroller known as Arduino UNO, together with the image processing algorithm run by MATLAB, the plan was to incorporate an obstacle prediction algorithm supported by two basic image processing techniques to estimate the traversability of the terrain. The deciding factor will be strongly influenced by the predefined threshold which relies on the wheel parameter set on the vehicle. Similarly, the authors in [4] had proposed a humanoid robot navigation algorithm that contains of image processing and optimization algorithm. They suggested that if all obstacles presented in an environment are contacted to the ground, the edge of the obstacle can then be identified as a point where the ground and obstacle collides. A reference pixel will then be selected for the utilization of corner detection to calculate the optical flow. Ergo, the optical flow will output a result for the authors to determine the moving speed of the obstacle presented on site. The information received through the monocular camera module will be subjected and the processed image will hence be optimized by the algorithm for the robot to move according to its optimal point.

As mentioned, UGVs are gradually becoming popular in the application for agricultural environments. Their applications are implemented to address shortage in human labor and even on the improvement for food safety throughout the production cycle. Other applications such as detection of animal fecal matter, detection of crop damage due to natural causes, survey on the growth of crops, and even detection of unwanted organisms is also commonly applied. In [5], Bonadies and Gadsden had implemented machine vision tactics to detect crop row edges and contours, leading to proper navigation and avoid damaging the crops. Firstly, lane detection algorithm was developed to allow robots to navigate through crop rows utilizing machine vision. The input will be interpreted by the algorithm to differentiate between crops and soils as compared to the authors in [4]. Once the position on the edges marked along the pave of the plantation field was processed, the center point/center of the row within the image will be determined. Figure 1 shows the row center determination illustration that can be implemented as the Region of Interest (ROI) for the UGV's mission in a plantation field. The purpose of having the ROI was to allow the UGV to focus on a specific setpoint calculated and located for it to visualize.

Two image processing methods were implemented for the detection of the edge for the crop rows. As the green objects such as trees, bushes, and plants are green in colour, the output pixels colour will tend to be brighter, a threshold adjustment for the brightness can be given to distinguish between the areas of grayscale, turning the input images into binary format. A global thresholding method was recommended to define all grey values exiting a threshold value as binary 1, also known as white. Vice-versa, the rest in black. Based on the research study, the lane detection approach can be further improved by applying filtering to its binary input images to remove all the existing noise, leading the threshold value to change against the ambient light in the field.

Journal of Physics: Conference Series **2523** (2023) 012030 doi:10.1088/1742-6596/2523/1/012030

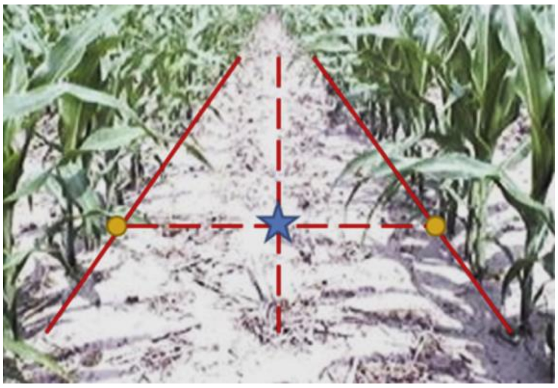

Figure 1. Calculated "starred shaped" center point and representation of predetermined setpoint

Likewise, vision-based application can also be enhanced to navigate a mobile platform autonomously. According to the research in [6] conducted by Minh Tran et al., a stereo camera was integrated to solve the rareness of navigation whereby unapplicable Global Position System (GPS) or low quality of GPS environment takes place. The camera was firstly set to cover a wide range of working area, followed by returning one frame for each sampling time. Within the time stem, the vision-based controller will program a path based on A* algorithm that generates a trajectory for the mobile platform to track down the existing obstacles. The tactic of retrieving the coordinates of the obstacles were done by partitioning the pre-process images into multiple segments. Later, removing the noises for each of the obstacles, before undergoing the A* path planning to meet its goal node. As stated in [5, 6], the image processing methods can be considered by further improving the ways of masking the objects of interest. Once the masking threshold values are determined, the pixels located on the camera's frame can estimate the distance between the detected obstacles to the UGV through monocular depth estimation.

On the contrary, with the employment of object detection and stereo cameras, the existing obstacles on site along with their sizes can be estimated through keypoints detection. In [7], four kinds of common vegetables such as cucumber, tomato, pepper, and eggplant were processed by detection networks to locate the six points for each vegetable. After going through the pre-trained keypoint detection networks, the proposed method can classify all four varieties of vegetable within a 60cm distance with an accurately predicted dimension. Moreover, convolutional neural network-based obstacle detection and obstacle classification method can be applied to visualize and predict images in the environment. It was studied by Ma et al., that a bidirectional feature pyramid network formed by the composition of ResNet and improved version of DenseNet can be further advanced by underlying more layers of details and high-level of strong semantic details in its architecture [8]. The proposed method was tested by a self-made dataset, open dataset, and Ablation experiments to evaluate its durability and consistency. Out of the other prospects such as mask RCNN, faster RCNN, improved faster RCNN and multi-scaling, the used of improved dense block to make the transmission of details more productive has awarded the proposed method as the best suited for autonomous navigation. But the number of labels and categories were only limited to the trained classes which is crucial to conduct more training towards various classification types to detect and recognize them.

Besides designing an artificial neural network with supervised image learning, Claudio et al. had conducted a preliminary study for classification and pattern recognition by utilizing the information retrieved by five SRF05 ultrasonic sensors [9]. Through the retrieving of returning waves, the ultrasonic sensors will detect the echo signal for the acquisitioning process. The Neural Network Toolbox presented in MATLAB was applied to design, train, and display onto the autonomous navigating vehicle. During the testing process, the type of obstacles met, and their respective positions were accurately being pointed out, leading to testing out its localization technique with the aid of multiple sensors. In contrary, the result obtained by the proposed method might be strongly affected by occluded obstacles. In model training, some factors that must be considered are the poor or abnormal lighting conditions, and the visibility of occluded objects [10]. These factors are concerning as uncontrollable lighting conditions and other variables are expected to differ in the plantation field. Occlusion must be mitigated as hidden objects might be present behind traversable obstacles, causing harm and damages toward the UGV's system and body. A baseline assessment for the state-of-the-art object detection models is suggested to be performed and compared for the much optimal outcome.

In this research project, a novel algorithm capable of detecting and classifying obstacles in a plantation field will yet to be created for the UGV to conduct its mission autonomously providing focusing on its safety through the additional implementation of a ROI bounded for its body. Ideally, the UGV will undergo a developed algorithm that holds the ability to detect common obstacles in a plantation field such leaves, rocks, and branches by segmenting them from its path. In addition, this research will mainly be focusing on the navigation of UGV in a uniformed plantation field such as cucumber, chilli, and eggplant farms. Different type of rover controller methods on autonomous UGV navigation in a plantation field will be critically reviewed. Followed by developing a plantation field environment-oriented classification and detection of obstacles algorithm that will be operating onboard of a companion computer known as NVIDIA Jetson Nano to assist on the UGV's mission. Lastly, implementing a Level 3 conditional autonomous path planning algorithm that can safely navigate the UGV within the plantation field. In the next section, it will provide a detailed description on the proposed obstacle classification and detection, and path planning algorithm to be implement for autonomous UGV navigation.

2. Methodology

Before diving into the specifics, Fig. 2 illustrates the general workflow for both the obstacle detection and path planning algorithm onboard of the Jetson Nano companion computer board. The experiment will solely be carried out on computer software containing computer vision and model training along with simulation that allows path planning to take place. Open-Source Computer Vision (OpenCV) is a Python Library that was chosen for the visualization and image processing to take place. It provides the UGV the proper vision to track down the existing obstacles presented on site. Moreover, TensorFlow, an open-source software library for machine learning was applied to process and load the pretrained models for the dataset training. It holds the ability to learn parameters that maps the input values to the desired output values, forming optimal parameters for machine. Lastly, MATLAB, a high-performance language for technical computing was used to simulate the proposed path planning algorithm.

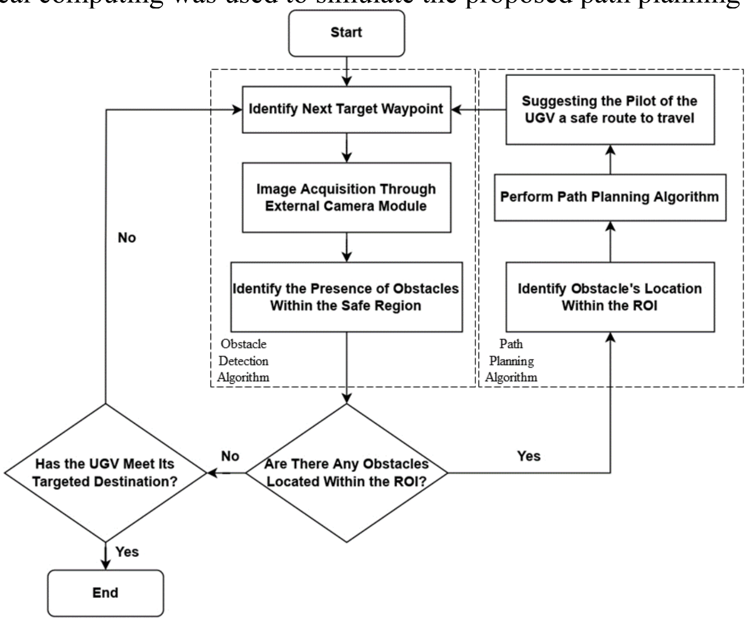

Figure 2. Proposed algorithm architecture

Journal of Physics: Conference Series **2523** (2023) 012030 doi:10.1088/1742-6596/2523/1/012030

The proposed method should be able to conduct instantaneous obstacle detection and classification for the UGV to navigate through the plantation field, while neglecting the obstacles that were not being set within the threshold values. Two components will be executed as obstacles are being detected, which are the obstacle detection and path planning algorithm respectively. The obstacle detection algorithm will be done by two approaches, one being the masking of images using Python OpenCV, while the other being training a custom object detector using TensorFlow object detection model. On the other hand, the path planning algorithm will be purely based on the implementation of a mathematical approach known as Trigonometric. The distances needed to perform the mathematics will be obtained through the pixels within the camera's frame.

2.1. Method

To operate the classification and detection of obstacles and path planning algorithm, a series of software and hardware were properly selected. The decisions made were crucial as it ensures the overall behaviour for the UGV will behave as accordance during the actual deployment on site. A summary of the software and hardware used were listed in Table 1 and Table 2 respectively. The details on the hardware used for the construction of UGV are omitted as it is not within the scope of the current study.

TABLE 1. Summary of software suites and libraries utilized

Software	Category	Functionality
Ubuntu	Operating System	Execute all software and libraries
Python	Programming Language	Implement and construct algorithms
OpenCV	Computer Vision Library	Image processing
TensorFlow	Machine Learning Library	Deep learning implementation and model training
CUDA	Programming Language for NVIDIA's Graphic Card	Enables the Graphics Processing Unit (GPU) to function
cuDNN	GPU-Accelerated Library for Deep Neural Networks	Provides highly tuned implementations for routines such as convolution, pooling, normalization, and activation layers
MATLAB	Numerical Simulation	Develop a path planning model

TABLE 2. Summary of hardware utilized

Software	Category	Functionality
Jetson Nano B01	Single-Board Computer	Execute neural networks such as image classification, object detection, and segmentation.
8MP IMX219 Low Distortion M12 Mount Camera Module	Vision and Image Capturing	Image acquisition along with providing real-time footages for the pilot

2.2. Obstacle Detection and Classification Algorithm

Masking is an image processing method to extract the ROI of an image. In OpenCV Python, it can be applied to create arbitrary masking shape using bitwise operations. This technique was implemented to filter out a range of green obstacles with a constant threshold value between 180 to 255 as green obstacles were researched to be commonly presented in a plantation field [11]. The detected obstacles that are classified under this image processing method will be focusing on obstacles that meets two conditions. Firstly, a ROI region of 480 × 200 will be set upon the 1280 × 720 camera frame. The purpose of having this ROI bounded within the frame was to keep the UGV running along the given safety region. As shown in Fig. 3, a measured distance of 1 meter between the UGV and the peak of bounding box was pre-calculated through the camera frame. If obstacles were detected within the bounded safety region, the masking method will then be executed. Secondly, after the masking of images, if the obstacles detected within the ROI has a density bigger than the given contour area value of 750, it will then prompt and alert the UGV regarding the upcoming obstacle.

Figure 3. ROI bounded within the camera frame

To achieve the proposed method, both the hardware and software needs to be integrated to form an integral part of the work. The masking algorithm will be constructed through Python and OpenCV library and the camera module will be mounted on the front part of the UGV at a constant height of 0.12m above ground level to capture and stream video continuously. As an initial step, the captured frame will be first converted into Hue Saturation Value (HSV) colour format as the Blue Green Red (BGR) image format is harder to be isolated. In the HSV colour format, the representation of colour, hue determines the colour we desire, saturation determines how intense the colour would be and value determines the lightness of the image. Based on, Fig. 4(a), it can be observed that the HSV image contains countless noise, ergo, further image filtering was needed to be done. Secondly, a range of green HSV colours ranging from [36, 50, 50] to [70, 255, 255] were mapped to match the green obstacles presented in the plantation field. Next, as shown in Fig. 4(b), the HSV image will undergo gaussian filter to reduce the existing noises running across it. Another filtering method to be implemented was the non-maximum suppression technique. This approached technique can aid on nullifying the remaining shadows, along with displaying the actual edges of the detected obstacles in a more enhanced manner. Moreover, the image will be further filtered by providing thresholding. With the above operations, the image's pixels will be compared to the given threshold value ranging from 180 to 255. If the pixel value is not within the threshold range, the respective pixel will be set to 0, whereas the remaining pixels falling under the threshold range will be set to maximum value. Furthermore, the thresholded image will be set according to the HSV colour range through the application of bitwise "AND" operation. Lastly, contour finding will be utilized to form a line joining for all the points inside the ROI, therefore the width of the obstacle can then be obtained for the following process. If the density of the obstacle meets a contour area of 750, it will then classify the obstacle as untraversable, forming a bounding box for the pilot's visualization. On the contrary, the remaining unbounded regions will be classified as traversable for the UGV to navigate. Figure 4(c) displays the outcome of an untraversable obstacle occurring within the ROI.

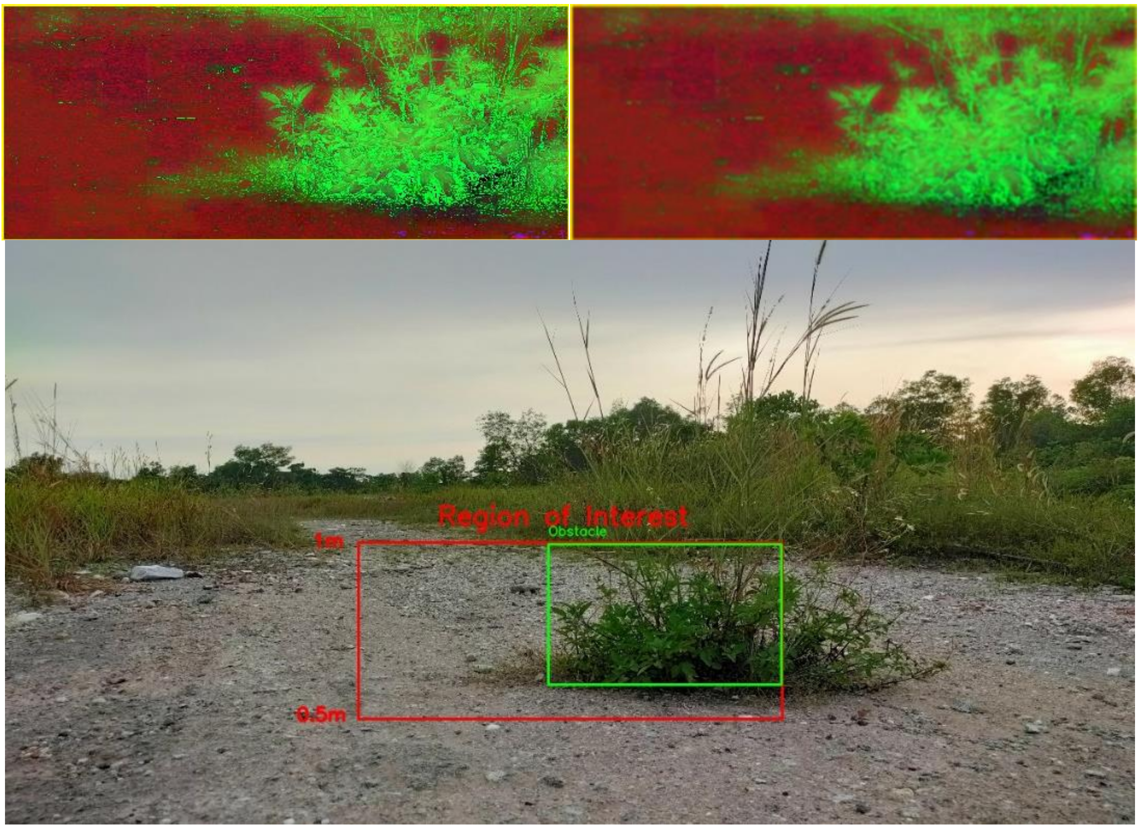

Figure 4. Image filtering operations applied to achieve image masking

Model training is a process of feeding a Machine Learning (ML) algorithm with the appropriate datasets to help extract and identify the elements being fed for it to learn on all the attributes involved. In this research, the supervised learning approach for classification and pattern recognition was implemented for the recognition process. A large third party of dataset containing leaves and rocks existing in a plantation field was inherited into the training process. The objects selected for the detection and classification to take part was due to its common presence on site [12]. By giving the UGV with such safety systems responsible of detecting and classifying the traversability of the respective obstacles, it can safeguard the UGV as well as preventing collisions with obstacles during the event of autonomous navigation. To result with a properly trained network, data augmentations were applied by using the available data and derive new images through them. In such a way where each newly cropped image was created consistently with the existing dataset for the machine to learn on a particular fragment out from it. This is a procedure where it increases the accuracy of the end model along with reducing the chances of overfitting and underfitting.

With the collected datasets, the stored images were first resized to a common 200 × 200-pixel image before any other image processing was done. Resizing was commerce due to the dataset varies in a range of size together with aiding the network to generalize better as it has a lesser amount of data to overfit. Next, the actual training will be carried out on a core platform and library for machine learning called TensorFlow. It will be used to build and compile the custom model for the training of network. In Fig. 5, it displays the architecture of the training model. Firstly, the input layer will take in the inputs and perform the calculations through its neurons. The number of neurons presented by the input layer was solely dependent on the dataset's output shape. Secondly, the output layer will be achieved through the calculations done via the neurons within the hidden layers of the neural network. The selection of hidden layers being inputted was crucial as the larger the number of hidden layers showing in a neural network, the more time it takes for the neural network to undergo the mathematical logics.

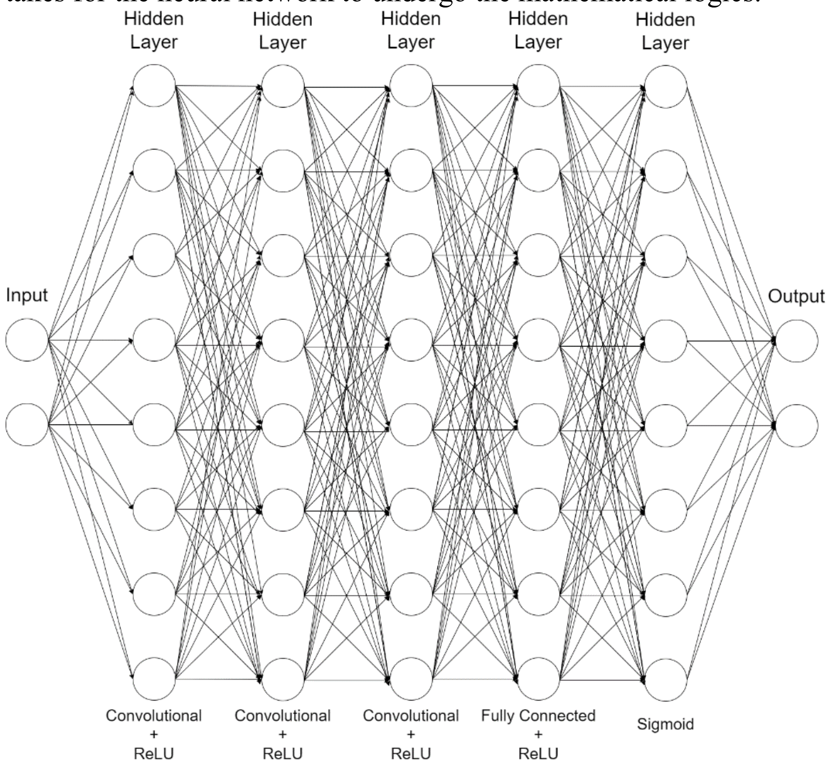

Figure 5. Architecture of training model

After carefully computing the training model, a total of 3 hidden layers were given between the input layers and output layers. The information stored in a form of arrays will first be flatten before passing on to the neural network by removing the interior arrays. In addition, the number of weight parameters of the first three layer was given with a unit of 16, 32, and 64 neurons consecutively. Subsequently, an activation function known as Rectified Linear Unit (RELU) was applied as it is a fast and efficient activation function that works best for this application. If the function returns value zero, it indicates the input value to be negative, vice versa. "Adam" optimizer was chosen due to it being a first-order adaptive optimizer. Entering to the network training section, the datasets will be separated into batches followed by running them in epochs along with validating the training process is feasible. The number of epochs will be tempered to find the most optimal rounds of training for the machine to undergo. The overall summary of the model training can be evaluated further in a table, observing the behaviours of accuracy and loss per epoch. Lastly, testing out the durability of the customize prediction model with the testing dataset prepared beforehand. The prediction model will result with the correct output by showing the output neuron with the highest probability count. Figure 6 illustrated the overall summary of the algorithm used to train the proposed technique.

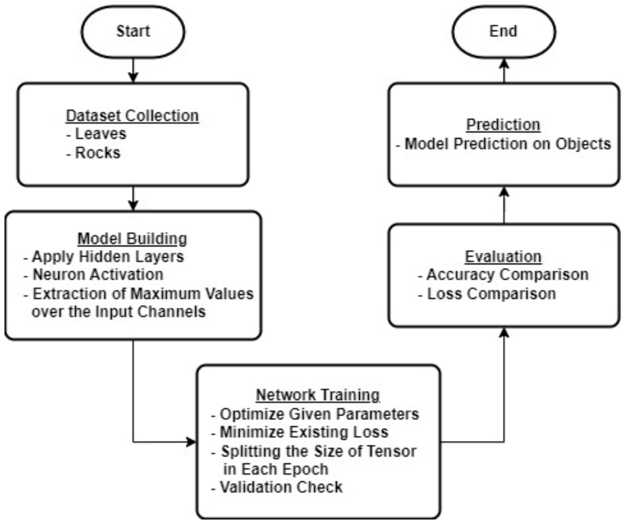

Figure 6. Overall flow of algorithm to train proposed method

2.3. Path Planning Algorithm

Path planning is a solution whereby it gives a feasible collision-free path that allows the UGV to get from one point to another. It is the key task for the UGV to conduct autonomous driving through the suggested command. In this research, a path planning algorithm was created to suggest the pilot behind the autonomous navigation UGV with conditional assistance. Moreover, the study was carried out in a computer software known as MATLAB. The details on how the UGV will be piloted by the pilot will be omitted as it is not within the scope of the current study. In addition to the mentioned statement, the UGV will first retrieve waypoint information from a GCS called Mission Planner. The given waypoint coordinates will provide the UGV with a better understanding on how the mission will be executed. Followed by utilizing the said object detection and classification model to track the untraversable obstacle within the ROI. If an obstacle was spotted in the ROI, the algorithm will then decide the most optimal path for the UGV to go about. Figure 7 displays the simulated environment in MATLAB whereby the UGV is represented by the white square labelled with the letter "R", an untraversable obstacle represented by the green square labelled with the word "Obstacle", a plotted waypoint for the UGV to conduct its autonomous navigation mission represented by the yellow square labelled with the prefix "WP", and two columns of green rectangles representing the uniform crops existing in the plantation field.

By implementing the obstacle detection and classification algorithm, the location of the obstacle can be presented through the captured frame. Therefore, the path planning algorithm can then acquire the upcoming information needed to bring the UGV over to a safe and collision free path. In Fig. 8, the red coloured dashed box represents the ROI displayed in the camera module. If untraversable obstacle is presented within the ROI, the UGV will then pause its mission, as it undergoes the path planning algorithm based on the trigonometry mathematical approach.

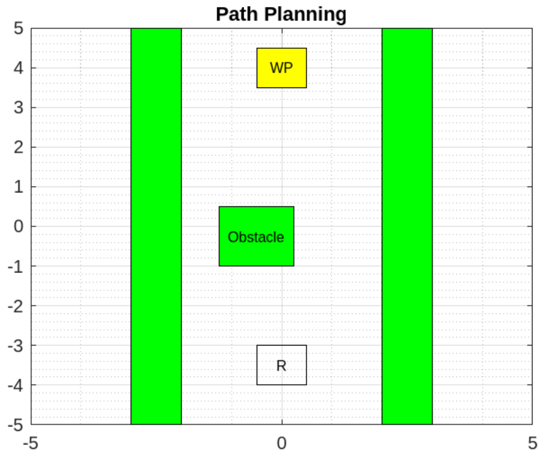

Figure 7. Simulated plantation field map

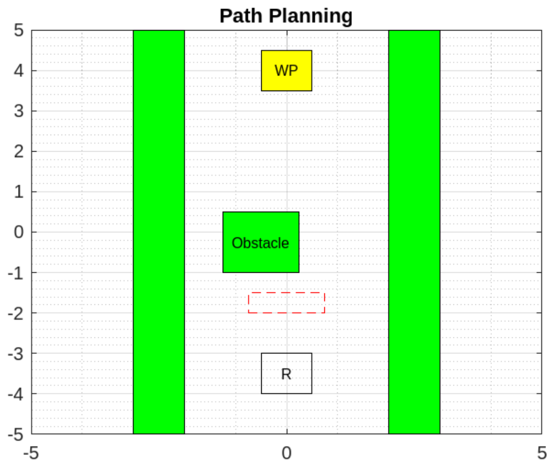

Figure 8. Simulated plantation field map with ROI displayed

3. Results and Discussion

In this section, a performance review on the classification and detection of obstacle algorithm along with the path planning concept will be discussed. Firstly, the results on the masking of images will be presented. Followed by the evaluation on the customized model prediction method along with some sample results collected. Lastly, the simulated path planning method will be tested and explained.

In Fig. 9, it was a sample scene taken by the camera module on site. Since the camera module was mounted over the front part of the rover in a constant height, the ROI will then be set in a bounding area ranging between 0.5 to 1 meter. Within the ROI, the mentioned approach was applied to track the obstacle existing in the region. If the density of the obstacle was strongly leaning onto the right area of the frame as shown in Fig. 9, it will then produce a suggested path for the UGV to be performed. This method was guaranteed as the path planning algorithm will ensure the suggested path will be feasible and traversable by the pilot.

Moreover, the customized training model conducted using TensorFlow was evaluated to examine the training and validation accuracy and loss. After going through the hidden layers in the built neural network, success is defined as the percentage for the accuracy of frames in a sequence having an accuracy value of 99.56% was achieved. Furthermore, the loss value had a drastic decreased from 9.58% to an excellent loss value of 1.49%. Note the validation accuracy had an exemption value with a precision of 99.54% as compared to the training set. A summary of the evaluation on the customized trained model was presented in Fig. 10.

Figure 9. Sample from the onboard camera of UGV in action

```
model.fit(x, y, batch_size=10, epochs=10, validation_data = validation_dataset)

Epoch 1/10
476/476 [==============================] - 36s 74ms/step - loss: 0.0958 - accuracy: 0.9639 - val_loss: 0.0373 - val_accuracy:
0.9939
Epoch 2/10
476/476 [==============================] - 30s 63ms/step - loss: 0.0467 - accuracy: 0.9865 - val_loss: 0.0381 - val_accuracy:
0.9863
Epoch 3/10
476/476 [==============================] - 30s 63ms/step - loss: 0.0215 - accuracy: 0.9943 - val_loss: 0.0127 - val_accuracy:
0.9954
Epoch 4/10
476/476 [==============================] - 30s 63ms/step - loss: 0.0338 - accuracy: 0.9901 - val_loss: 0.0546 - val_accuracy:
0.9909
Epoch 5/10
476/476 [==============================] - 30s 64ms/step - loss: 0.0184 - accuracy: 0.9939 - val_loss: 0.0275 - val_accuracy:
0.9939
Epoch 6/10
476/476 [==============================] - 30s 64ms/step - loss: 0.0228 - accuracy: 0.9918 - val_loss: 0.0484 - val_accuracy:
0.9863
Epoch 7/10
476/476 [==============================] - 30s 63ms/step - loss: 0.0104 - accuracy: 0.9964 - val_loss: 0.1070 - val_accuracy:
0.9636
Epoch 8/10
476/476 [==============================] - 30s 64ms/step - loss: 0.0167 - accuracy: 0.9947 - val_loss: 0.0114 - val_accuracy:
0.9970
Epoch 9/10
476/476 [==============================] - 30s 62ms/step - loss: 0.0141 - accuracy: 0.9962 - val_loss: 0.0044 - val_accuracy:
0.9970
Epoch 10/10
476/476 [==============================] - 30s 62ms/step - loss: 0.0149 - accuracy: 0.9956 - val_loss: 0.0160 - val_accuracy:
0.9954

<keras.callbacks.History at 0x2995fb262f0>
```

Figure 10. Evaluation scheme of the customized trained model

Once the model was trained and evaluated, it was brought out for the prediction to be conducted. In Fig. 11, it displays some samples on test images used to experiment on the trained model. As illustrated, the predicted model in TensorFlow was able to output with the precise outcome bounded in red for the images being fed into the system. Therefore, the built model can further be operated for the detection of obstacles on site. According to the existing density of the respective obstacles, it will be used to classified on whether the obstacle is traversable or untraversable by the UGV. This will ensure the UGV is protected at all causes and time, preventing damages or collision from occurring.

During the simulation, when the UGV progressively travel towards the waypoint, obstacles will be met. As shown in Fig. 12, the UGV will then activate the path planning algorithm to seek for available spaces for the UGV to travel through.

The distance between the UGV and detected obstacle with bounding box will be highlighted through the camera's frame. Then, the distance between the midpoint of the frame and the width of the bounding box overlapped will be captured. By collecting both the distance measurement, it can then be fed into Equation 3 to seek for the theta angle needed to be yaw by the UGV to meet the safe distance. Figure 13 shows the UGV collecting the distances required for the calculation to take place. The distances were solely based on the bounding boxes dimension occurring during obstacle detection.

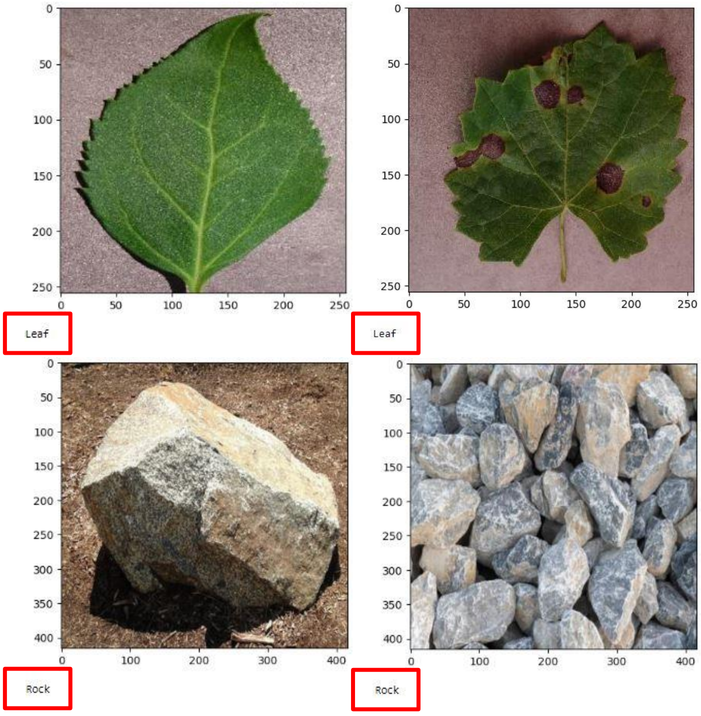

Figure 11. Results from the prediction outcome

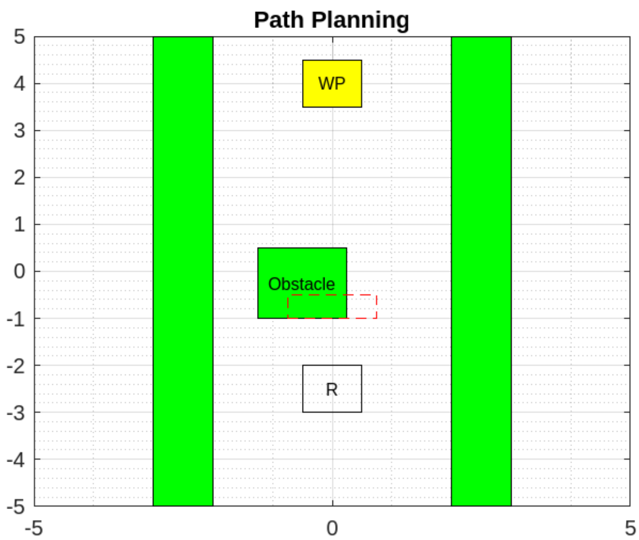

Figure 12. UGV detected an untraversable obstacle

Once the theta value was calculated, the UGV yawed according to the given value to ensure the UGV can continue its mission within a safe region was shown in Fig. 14(a). Furthermore, the distance needed to be travelled forward was mapped to a distance calculated beforehand according to the angle needed to be yawed. In Fig. 14(b), it displays the scenario of the UGV travelling onto the safe region.

The path planning algorithm will be further implemented in the future study, as the current scenario was to be simulate a safety region before providing the pilot with a safe path.

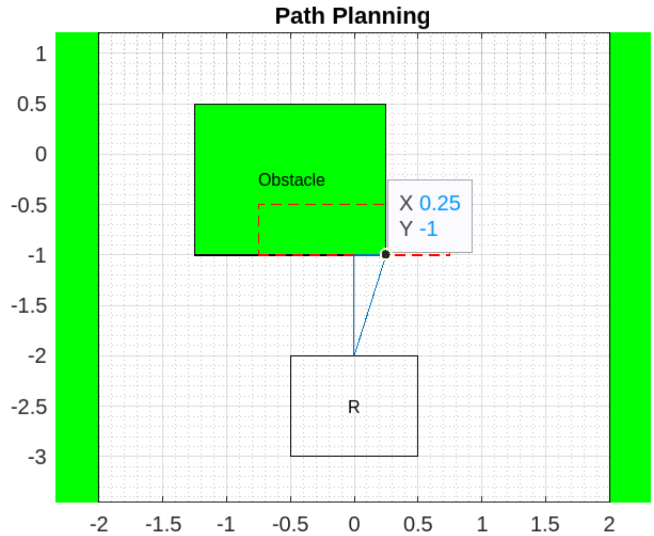

Figure 13. UGV collecting the distance measurement through the pixels dimension

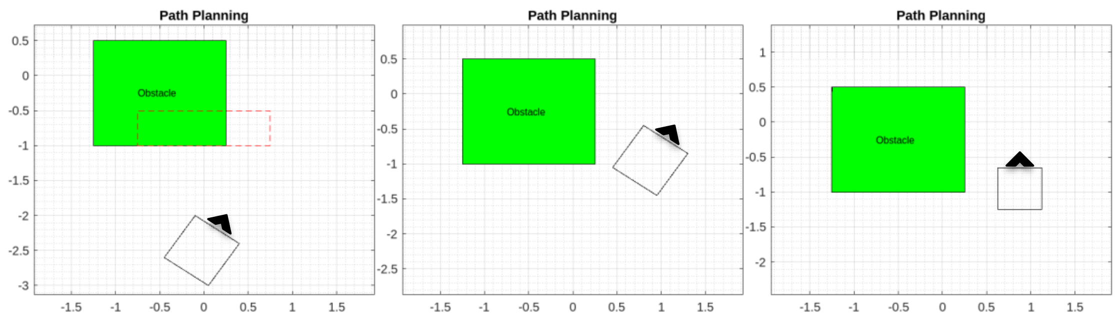

Figure 14. Behaviour of UGV as it simulates to seek for a safety region

4. Conclusion

Through this study, a classification and detection of obstacle and a path planning algorithm was proposed. This proposed algorithm was designed from scratch based on its workability, practicality, and productivity as the UGV conducts a mission in a said range of plantation fields. It can be concluded that the objectives of the research study were fulfilled. After undergoing data training, the customized leaf and rock datasets were also successfully being implemented to work along the proposed object detection algorithm. It can be certain that the density of one object's pixel size can determine the next step of the UGV's action and therefore, the used of the selected, Jetson Nano 8MP IMX 219 camera module plays a crucial role with the capturing of real time images. Furthermore, the outcome of the model prediction on leaves and rocks can be applied for the UGV to catch out the unwanted untraversable obstacles prevailing its path. With the use of the image masking technique, along with the model prediction technique, the existing obstacles within the ROI will be boxed out. In addition, the UGV will conduct the suggested path planning algorithm through the implementation of the trigonometrical mathematic logic that will then prompt the pilot with a suggested route to traverse. With this technology, the UGV can conduct agricultural related issues such as watering, spraying pesticides, crop checking and even act as a payload carrier on site autonomously.

As per the current outcome delivered by the author, many components can be replaced and improved upon. A new and improved training and validation dataset can be implemented to the future for a much ideal and elevated performance on the obstacle detection process. Moreover, the construction of the neural network architecture can be tempered to collect a newly improvised training model by feeding the network with other appropriate and ideal hidden layers. Furthermore, stepping up the UGV's automation level by giving the UGV more autonomy and decision-making authority to develop a new and improved path planning system. Lastly, a remote base station with the capability to provide the UGV a conditional autonomous respond by alerting the pilot behind the station can be proposed to be constructed off site. Ergo, conducting the navigation mission wirelessly can lessen the need of workers to physically be on site as the harvesting and other properties can properly be executed.

References

[1] Cao, Z. and Sadigh, D., 2021. Learning from imperfect demonstrations from agents with varying dynamics. *IEEE Robotics and Automation Letters*, **6(3)**, pp.5231-5238.

[2] Kulkarni, A.A., Dhanush, P., Chetan, B.S., Gowda, C.T. and Shrivastava, P.K., 2020, February. Applications of automation and robotics in agriculture industries; a review. In *IOP Conference Series: Materials Science and Engineering* (Vol. 748, No. 1, p. 012002). IOP Publishing.

[3] Reddy, G.V.P. and Nagaraja, S.R., 2016, December. Image based obstacle detection and path planning for solar powered autonomous vehicle. In *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)* (pp. 1-7). IEEE.

[4] Han, Y.J., Kim, I.S. and Hong, Y.D., 2018. Optimization-based humanoid robot navigation using monocular camera within indoor environment. *ETRI Journal*, **40(4),** pp.446-457.

[5] Bonadies, S. and Gadsden, S.A., 2019. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. *Engineering in Agriculture, Environment and Food*, **12(1),** pp.24-31.

[6] Zhang, L., Deng, F., Chen, J., Bi, Y., Phang, S.K. and Chen, X., 2018. Trajectory planning for improving vision-based target geolocation performance using a quad-rotor UAV. *IEEE Transactions on Aerospace and Electronic Systems***, 55(5),** pp.2382-2394.

[7] Zheng, B., Sun, G., Meng, Z. and Nan, R., 2022. Vegetable Size Measurement Based on Stereo Camera and Keypoints Detection. *Sensors***, 22(4),** p.1617.

[8] Ma, L.Y., Xie, W. and Huang, H.B., 2019. Convolutional neural network based obstacle detection for unmanned surface vehicle. *Mathematical biosciences and engineering: MBE*, **17(1),** pp.845-861.

[9] De Simone, M.C., Rivera, Z.B. and Guida, D., 2018. Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors. *Machines***, 6(2),** p.18.

[10] Jefferson, D.I., 2022. *Baseline Assessment of Object Detection Models on Partially Occluded Objects*. DEVCOM Army Research Laboratory.

[11] Campos Silvestre, Y., 2018. Computer vision techniques for greenness identification and obstacle detection in maize fields. *Ene*, **11**, p.31.

[12] Gonzalez-De-Santos, P., Fernández, R., Sepúlveda, D., Navas, E. and Armada, M., 2020. Unmanned ground vehicles for smart farms. *Agron.-Clim. Chang. Food Secur***,** **6**, p.73.