ORIGINAL ARTICLE

A Novel Automated Tree Detection System using Faster R-CNN

Jayson Piquero[1,*], Edwin Sybingco[1], Alvin Chua[2], Marc Say[1], Clarisse Crespo[1], Reginald Rivera[1], Ma. Antonette Roque[1], Leonard Ambata[1]

[1]*De La Salle University. Department of Electronics and Communications Engineering*

[2]*De La Salle University. Department of Mechanical Engineering*

*Corresponding author: jayson_piquero@dlsu.edu.ph

Received: 23rd August 2019

Accepted: 17th May 2020

OPEN ACCESS

Abstract: This paper introduces an approach to tree detection using mapped aerial images from drones. Tree detection is achieved by training a Faster Region-Based Convolutional Neural Network using Tensorflow Object Detection API. A Color-Based tree detector is also added to further filter out undesired detections such as tree shadows. Aerial images obtained from drones are mapped using Web Open Drone Map, an open-source drone mapping software. This paper aims to achieve a low-cost approach to disaster assessment using tree detection and drone images. The implemented system is able to detect trees and yielded an average detection rate accuracy of 85.63% and 95.14% for stitched maps and unstitched images respectively.

Keywords: drone, mapping, webodm, faster r-cnn

Introduction

Drones have been used for multiple applications such as surveying, agriculture, photography, and many more. This led to an increase in the study and research about drones. There are numerous researches on drones used for surveying and analysis due to the ease of data gathering.

The study in [1], similar to this paper, aims to detect trees using unmanned aerial vehicles. The study detects trees using point clouds obtained from LiDAR data. Analysis of 3D data gives an accurate detection, however it relies on Agisoft Software which is expensive.

The study in [2] is also designed to detect objects using mapped aerial images. This is also a low-cost surveying method for post-disaster assessment. Despite the low-cost stitching method, the stitched maps have distortions and are dependent on the lighting. The object detection implemented was single-shot detector which is a fast detector model but not the most accurate.

The work in [3] is a low-cost drone system used for analyzing elevation. Using a LiDAR lite V3 sensor and Arduino Uno, a surveying method is achieved.

The implementation in [4] also involves drones that takes aerial images. The study focuses on determining vegetation of areas for agricultural purposes. Similarly, this study also made use of Web Open Drone Map to stitch the aerial images taken. However, the study is only used to determine areas of vegetation instead of counting trees one by one.

In [5] and [6], a Viola-Jones classifier was trained in order to detect hydromorphological features around rives such as roads, shore, trees, and roads. Similar to this study, these applications can be used for disaster assessment and surveying. However, these are not applied to stitched maps which lessens the coverage of the detection.

Additionally, the work in [7] was also a drone equipped with camera for surveying, however instead of taking pictures for mapping, this uses video stream. Video streams have lower resolution and are unnecessary for mapping which is why this method is not efficient for object detection.

Other drone studies were also conducted by applying a more accurate attitude controller such as Sliding Mode Controller in [8] and Fuzzy GS-PID Controller in [9] to improve drone performance while carrying payloads. Other frame types, experimental setups, and flight controllers were also introduced for improved applications such as obtaining mapping images as described in [10] and [11].

This paper introduces a low-cost method of accurately mapping an area using Web Open Drone Map and detecting the number of trees in the area with a trained Faster R-CNN model. The model is trained using Tensorflow, an open-source library intended for machine learning. Color filtering with OpenCV libraries is also added to increase the accuracy of the detection. With the functionalities of this system, a novel method of disaster assessment, surveying, and agricultural mapping is achieved.

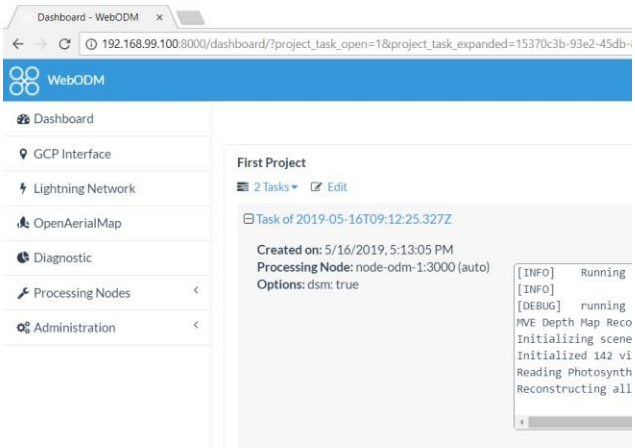
Design Considerations

Web Open Drone Map

Web Open Drone Map (WebODM) is an open-source software that allows the generation of maps, point clouds, digital elevation maps, and 3D models from drone images. Because WebODM is free compared to expensive mapping software such as Pix4D, this was the mapping software chosen in this study as the aim is to develop a low-cost mapping and tree detection algorithm with drone images despite the slower performance.

This paper uses WebODM for 2D mapping of aerial images. The aerial images have to be geotagged in order to produce orthomosaic maps. These geotagged images were taken from different drone frames such DJ Phantom 3, DJI Phantom 4, and a Skyhunter Fixed-Wing equipped with Pixhawk Flight Controller. WebODM is installed in Windows using Docker Quickstart Terminal and Virtualbox. WebODM can be opened by running the virtual machine through docker and visiting the dedicated IP Address and port using your internet browser.

Figure 1 shows the WebODM application. In this application, the user can import the aerial images from drones, and the stitching progress can be seen. Table 1 shows the duration of each mapping task with the corresponding number of images and the drone frame types used to obtain the images.



192.168.99.100:8000/dashboard/?project_task_open=1&project_task_expanded=15370c3b-93e2-45db-

Task of 2019-05-16T09:12:25.327Z

Created on: 5/16/2019, 5:13:05 PM
Processing Node: node-odm-1:3000 (auto)
Options: dsm: true

```
[INFO]    Running
[INFO]
[DEBUG]   running
MVE Depth Map Reco
Initializing scene
Initialized 142 vi
Reading Photosynth
Reconstructing all
```

Figure 1. WebODM WebApp

Table 1. Stitching time of images

Image Set	Number of Images	Duration	Drone Frame Type
1	35	1hr 22mins	DJI Phantom 4
2	25	1hr 11mins	Skyhunter
3	85	7hrs 8mins	DJI Phantom 3
4	55	2hrs 30mins	DJI Phantom 3
5	74	4hrs 57mins	DJI Phantom 3
6	60	2hrs 57mins	DJI Phantom 3
7	55	2hrs 46mins	DJI Phantom 3
8	54	2hrs 22mins	DJI Phantom 3
9	79	6hrs 30mins	DJI Phantom 3
10	185	9hrs 29mins	DJI Phantom 3
11	239	12hrs 29mins	DJI Phantom 3

Table 1 shows the duration of all the sets of images for mapping, however this does not completely define the performance of WebODM as it also depends on the specifications of the computer.

Figure 2 is the orthomosaic output of WebODM using Image Set 2 from Table 1. This was mapped using a Skyhunter frame. The images were geotagged using Mission Planner. The other Image Sets from Table 1 were taken using DJI Phantom 3 and 4 which are already geotagged. Image Sets 3 to 11 are taken from [2] and are applied to WebODM. These mapped images are then used for object detection.

R-CNN, Fast R-CNN, and Faster R-CNN

The tree detection is achieved by training 914 images of images taken by drones with the Convolutional Neural Network (CNN) architecture. These images are all annotated before training. Annotation is used to determine the locations of all the trees in each image. This is done by using the tool LabelImg.

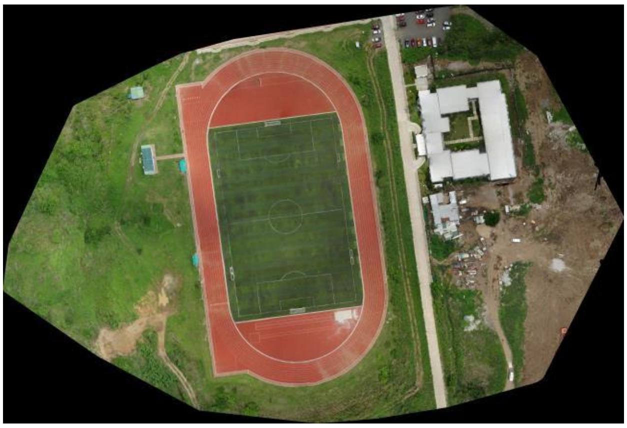


Figure 2. Image set 2 mapped with WebODM

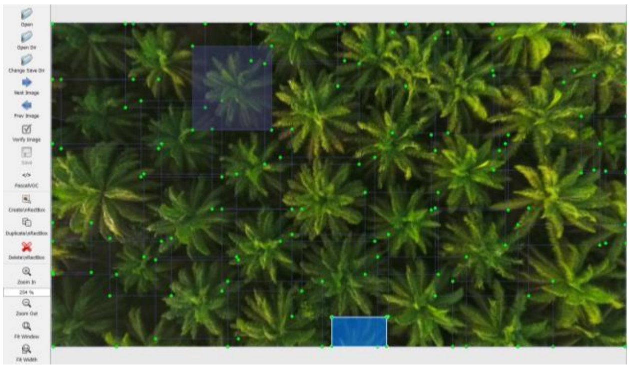


Figure 3. Annotation using LabelImg

Figure 3 shows how all the images were manually annotated to be used for the object detection. The training images were all obtained from Google Images. This paper uses Tensorflow, an open-source platform for machine learning, to train the CNN architecture.

The CNN used for this paper is the Faster Region-Based Convolutional Neural Network (Faster R-CNN). This was chosen due to its high accuracy and efficiency [12]. Since this is post-processing and does not need to be run in real-time, the speed of processing is not prioritized.

Faster R-CNN is an improvement of Fast R-CNN and R-CNN. The Region-Based Convolutional Neural Network algorithm uses a method called selective search to extract boxes from an image that could contain objects.

Selective search identifies patterns in an image [13]. Such patterns considered are colors, textures, size, and edges as seen on Figure 4. Similar regions are combined to form bigger regions with the numbers indicating the confidence of detection. These regions are considered as the objects in the image and are classified to their respective object types.

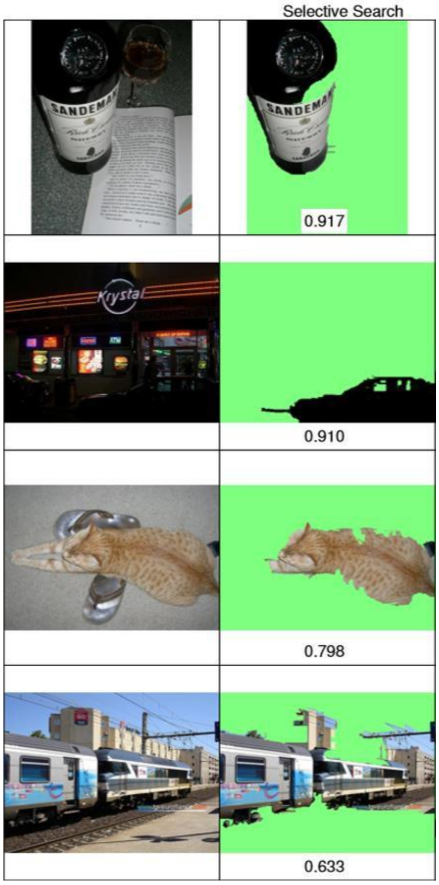


Figure 4. Selective Search [13]

These regions are reshaped and passed onto a CNN to extract features and to properly classify each region to their respective classes. For each region, a bounding box is predicted to properly locate the object in the image. These steps are illustrated in Figure 5.

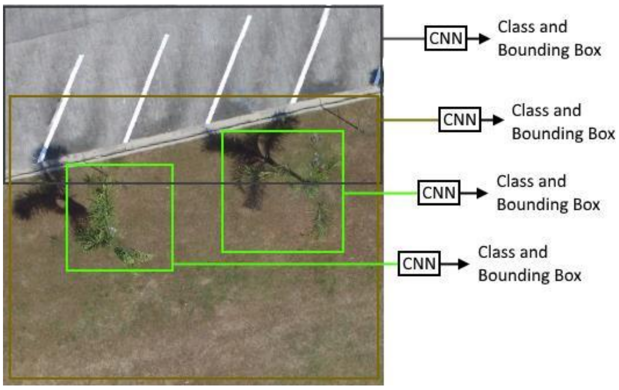


Figure 5. R-CNN Object Detection

Figure 5 shows a sample image where regions are chosen using selective search and are passed to a CNN for feature extraction. These features are then classified to their types such as trees, roads, ground. However, in this paper, the Faster R-CNN is trained to detect only trees.

Because RCNN extracts roughly 2000 regions per image using selective search and makes use of three models (CNN for feature extraction, Linear SVM for object classifier, and regression for creation of bounding boxes), the CNN model becomes slow. The slow detection of RCNN resulted in the development of Fast R-CNN [14].

Fast R-CNN, instead of taking roughly 2000 regions per image and sending each region to a CNN for feature extraction, sends the whole image to a CNN to obtain a feature map of the whole image. Selective search is used to divide the feature map into regions of interest. These regions are then passed to a fully-connected layer containing softmax and linear regression layers. The softmax layer and the linear regression layer are used to classify the regions and return their bounding boxes respectively.

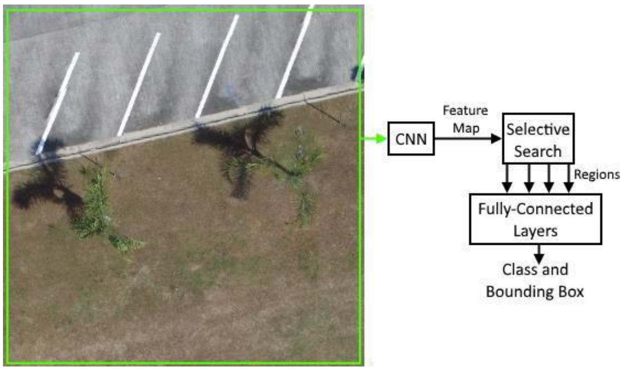


Figure 6. Fast R-CNN Object Detection

Figure 6 shows how objects are detected using Fast R-CNN. Compared to R-CNN which initially uses selective search and gets 2000 regions, Fast R-CNN sends the whole image first to a CNN and obtains a feature map. This saves a lot of computational time compared to sending 2000 regions to CNNs. However, R-CNN also makes use of selective search which also consumes time during detection.

Faster R-CNN is a modification of R-CNN as presented in [6]. Similar to Fast R-CNN, Faster R-CNN passes the input image to a CNN and obtains a feature map. The distinguishing factor of Faster R-CNN from Fast R-CNN is the use of Region Proposal Network (RPN) over selective search. The feature map is passed to the RPN which returns the object proposals along with the probabilities that the selected proposals are objects. The output from the RPN and the feature map are all passed on to a fully-connected layer which has a softmax layer and a linear regression layer. Similar to Fast R-CNN, the softmax layer and the linear regression layer are used to classify the regions and return their bounding boxes respectively.

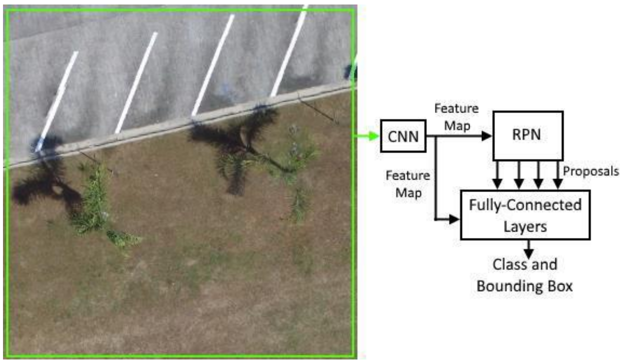


Figure 7. Faster R-CNN Object Detection

As can be seen on Figure 7, RPN replaces selective search in Fast R-CNN. The RPN and the fully-connected layers are trained for effective detection. The 914 aerial images of trees were used to train this model. Since only trees were trained, only one class can be detected by this model. This model is then used for tree detection in mapped aerial images.

Color Filtering

Color filtering is an optional feature of this system which can help filter out the false detections from Faster R-CNN. Such false detections include tree-like figures such as shadows of trees. This can be demonstrated on Figure 8 which was obtained from [15].

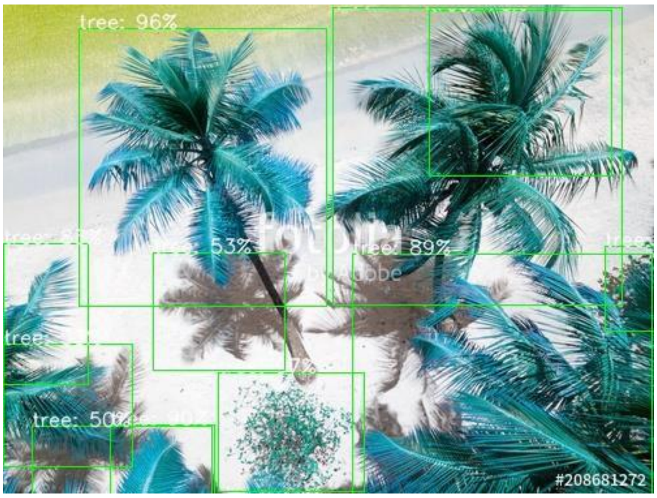
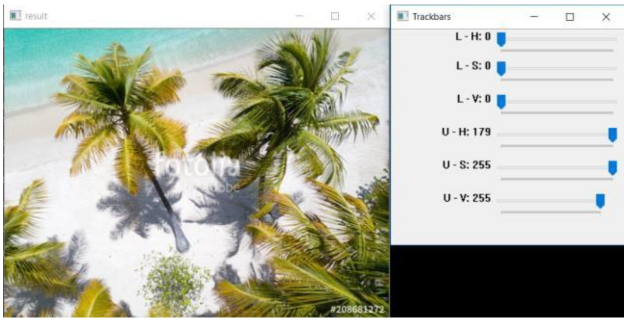


Figure 8. Sample Object Detection using Faster R-CNN

As seen on Figure 8, one of the shadows of the trees is included in the detection with a score of 53%. Because the model was trained with images of trees that are not only green, the detection sees the shadow of the trees as actual trees.

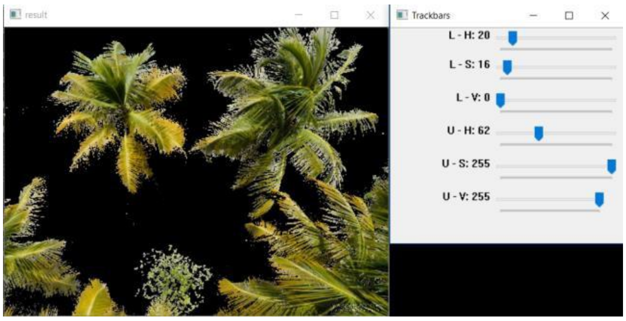
To counter this, this paper uses an additional color filtering scheme that allows a user to manually pick a range of color acceptable using OpenCV libraries. Colors outside the desired range are remove. This means that the false detections due to shadows are removed from the image.



L - H: 0
L - S: 0
L - V: 0
U - H: 179
U - S: 255
U - V: 255

Figure 9. Sample image for color filtering

Figure 9 shows an image taken from Google Images and the interface for the object detection. Trackbars will appear that allow users to choose the upper and lower limits of color in Hue-Saturation-Value (HSV) format. Pixels of the image that are outside the upper and lower HSV limits are filtered out.



L - H: 20
L - S: 16
L - V: 0
U - H: 62
U - S: 255
U - V: 255

Figure 10. Filtered color of sample image

Figure 10 shows the sample image from Figure 8 applied with the color filtering. The upper and lower HSV values chosen for this particular image are 62 to 20, 255 to 16, and 255 to 0 respectively. Further morphological transformations are done such as erosion and dilution to the filtered image to reduce noise.

Figure 11 shows the output of the object detection with application of color filtering. Compared to Figure 8, this removes the shadow that was detected as an actual tree. This helps improve the accuracy of the detection system. Please see the detected images to compare.

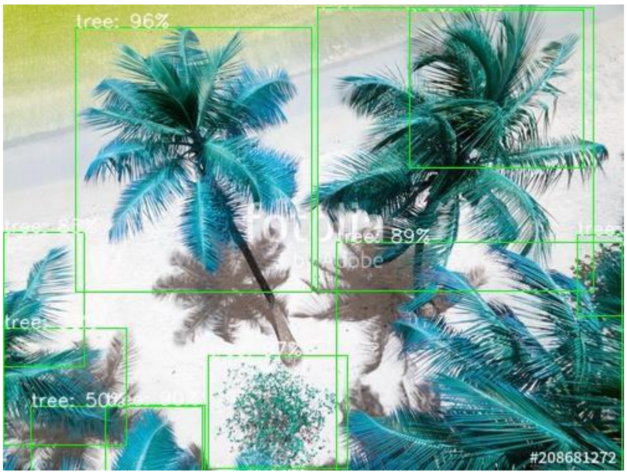


Figure 11. Sample Image Object Detection with Color Filtering

Methodology

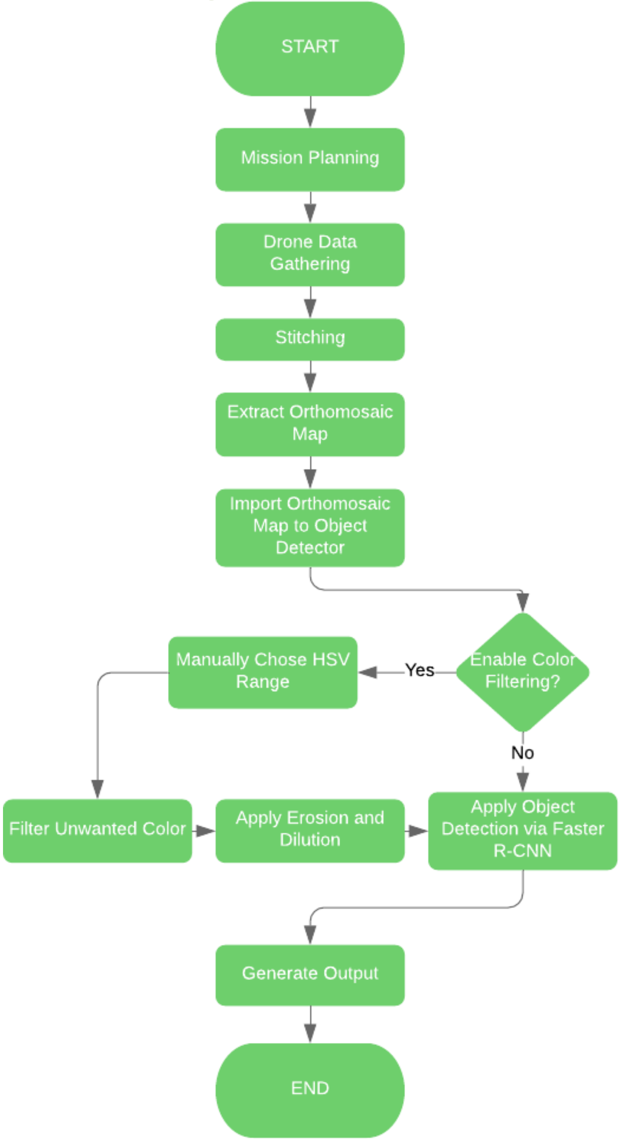
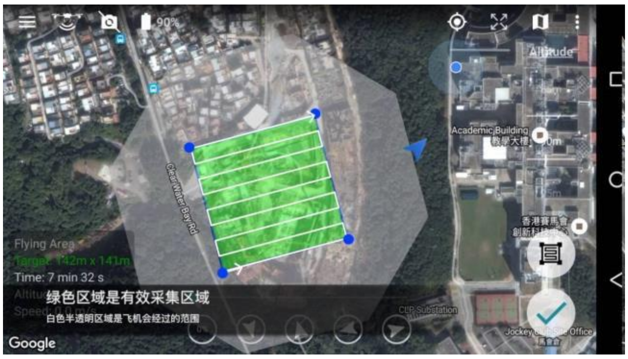


Figure 12. System Flowchart

As seen on Figure 12, the system is started by mission planning the drone. Since there were 3 drone frames used in this paper, different methods of mission planning were used. To plan the missions with the DJI Phantom 3, an iOS app called Altizure for DJI is used to plot waypoints and trigger the camera.

Figure 13, taken from [16], shows how the mission of DJI Phantom 3 is planned. This allows the user to vary the altitude, waypoints, and waypoints. These waypoints are then imported to the DJI Phantom 3.



绿色区域是有效采集区域
白色半透明区域是飞机会经过的范围

Figure 13. Altizure for DJI Mission Planning

Similar to DJI Phantom 4, an iOS app called Map Pilot for DJI Phantom 4 is used for mission planning. Figure 14, taken from [17], shows the interface of the app. Similarly, missions are set with the desired altitude and number of images.

For the skyhunter fixed-wing frame, Pixhawk flight controller was used as the autopilot. To create missions with this autopilot, a software called Mission Planner was used. Figure 15, taken from [18], shows the waypoints and camera triggers for data gathering.

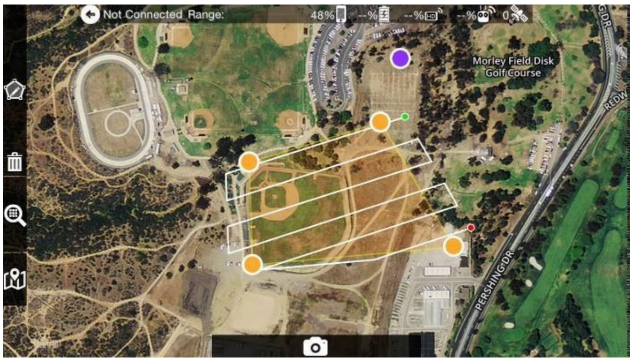


Figure 14. Map Pilot for DJI Mission Planning

Once the images are extracted from the drone, these images are imported to WebODM for stitching. As discussed before, an orthomosaic map, 3D point clouds, and elevation model can be generated with WebODM, but in this paper, only orthomosaic maps will be used as the detection model is only for 2D images.

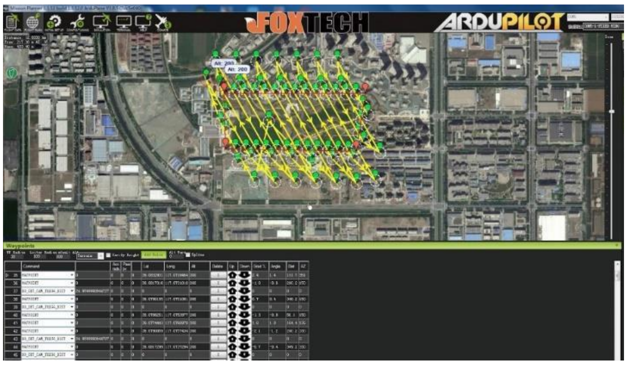


Figure 15. Mission Planner

The extracted orthomosaic map will be manually inputted to the object detector. If color filtering is enabled, the user is able to choose the desired range of color in HSV format as discussed in the Color Filtering section. However, this feature is optional and can be skipped by the user.

Once color filtering is done or is skipped, object detection via Faster R-CNN is used. As in the discussion of Faster R-CNN, the bounding box and the corresponding object class is outputted from the model. Bounding boxes that are outside the desired color area is not included in the detection. Additionally, objects detected with Faster R-CNN that have objectness scores below a certain threshold (typically 50%) are removed. With this, object detection in orthomosaic maps is achieved.

Data and Results

2D Mapping

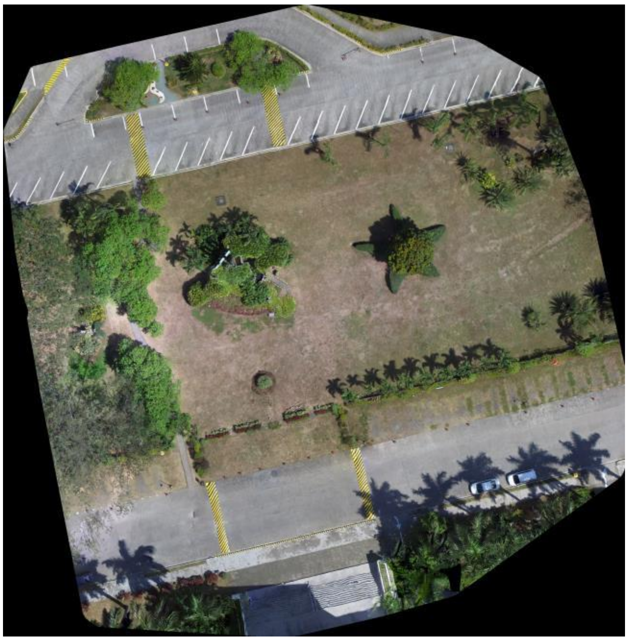


Figure 16. Stitched Map of Image Set 7

Figure 16 shows the stitched map from Image Set 7 as described in Table 1. This is inputted to the object detector for tree counting. Both color filtering and tree detection were used to produce the correct number of bounding boxes.

Object Detection on Stitched Maps

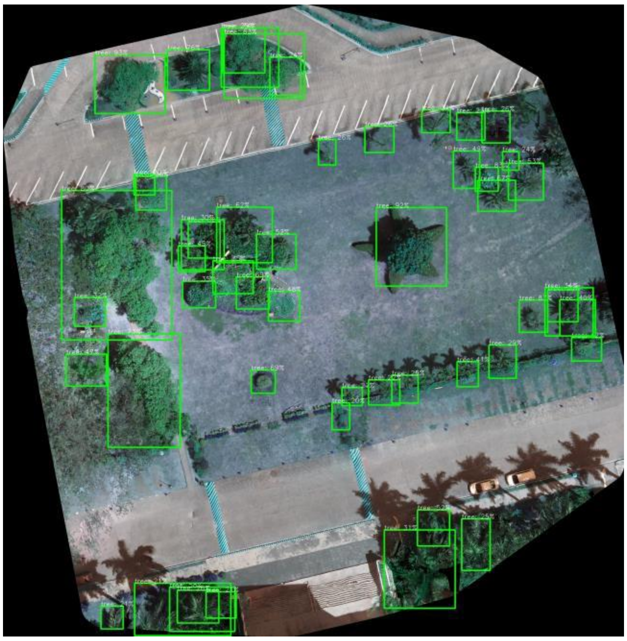


Figure 17. Object Detection of Stitched Map of Image Set 7

Figure 17 shows the bounding boxes around the stitched map. Their classification which is “tree” is displayed along with each objectness score. The user is able to change the threshold of the objectness score to any value desired (typically 50%).

This model is tested on all 11 image sets as listed on Table 1. This data was used to compute the accuracy of the model as shown in Table 2.

Table 2. Summary of Detected Objects on Stitched Images

Image Set	No. of Trees Detected	No. of Actual Trees
1	9	12
2	11	14
3	10	9
4	5	7
5	6	6
6	36	45
7	52	59
8	48	50
9	18	20
10	22	23
11	18	23

Table 2 shows the number of detected trees and the number of actual trees. The percentage error can be computed by the average of the absolute difference of the number of actual trees and the number of detected trees divided by the number of actual trees. The percentage error of the detection rate taken from this data is 14.37%.

Object Detection on Unstitched Images

Similarly, this approach was also used on aerial images that were not stitched for further redundancy testing.

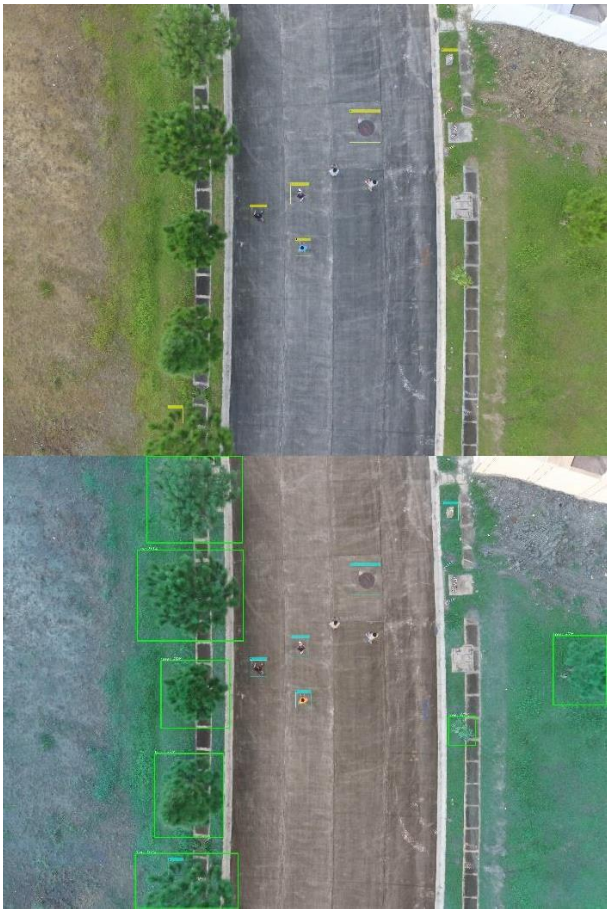


Figure 18. Object Detection on Unstitched Image [16]

The upper image on Figure 18 is an image from [16] that shows an aerial image. This was inputted to the object detector model, and the lower image is the output. As can be seen on Figure 18, 7 out of 7 trees were detected. In example, the model achieved 100% detection accuracy. This is also tested on other singular unstitched images as seen on Table 3.

Table 3 shows the number of trees that are detected by the model with the unstitched images. From this table, an accuracy of 95.14% for the detection rate is obtained. This shows a higher accuracy for smaller and unstitched images.

Table 3. Summary of Detected Objects on Stitched Images

Image #	No. of Trees Detected	No. of Actual Trees
1	5	5
2	7	7
3	9	9
4	1	1
5	18	21
6	5	5
7	9	8
8	12	12
9	15	17
10	9	10

Conclusion

The implemented system was successful in creating orthomosaic maps with aerial drone images using WebODM and detecting the trees using the trained Faster R-CNN model. Despite the slower stitching of WebODM compared to ready-made software, it is desirable to use WebODM especially for low-cost applications.

Because the trained detection model arrived at an average detection rate accuracy of 85.63% for stitched maps and 95.14% for unstitched images, the system was successful in detecting the desired objects. The accuracy in unstitched images showed higher values than the stitched maps. This may be because the images used for training the Faster R-CNN model were unstitched images. The color detection also showed successful filtering as it removes shadows and other unwanted detections from the images.

For further improvement, stitched maps may also be added to the training data of the detection model. In the application of disaster assessment, other objects may also be considered for training detection models such as structures and roads. Additionally, point clouds may be used for object detection schemes since WebODM also produces point clouds and elevation models.

Acknowledgements

The authors would like to acknowledge the Department of Science and Technology – PCIEERD with project number 04254 and the University Research Coordination Office in De La Salle University for funding and supporting this research.

References

[1] M. Mohan, C. Silva, C. Klauberg, P. Jat, G. Catts, A. Cardil, A. Hudak, and M. Dia, "Individual Tree Detection from Unmanned Aerial Vehicle (UAV) Derived Canopy Height Model in an Open Canopy Mixed Conifer Forest", 2017. https://doi.org/10.3390/f8090340

[2] L. Ambata, A. Chavez, N. Devnani, J. Ingeniero, A. Paggabao, "LEAD: A LiDAR-based Three-Dimensional Mapping System for Environment and Object Detection on an Unmanned Aerial Vehicle", B.S. thesis, De La Salle University, Manila, Philippines, 2019. https://doi.org/10.1109/HNICEM48295.2019.9072891

[3] W. Metzler, D. Pinson, A. Hendrickson, R. Xu, and J. Henriques, "Low-Cost Drone System for Analyzing Elevation", 2018. https://doi.org/10.1109/SIEDS.2018.8374732

[4] J. Barnetson, S. Phinn, and P. Scarth, "Mapping woody vegetation cover across Australia's arid rangelands: Utilising a machine-learning classification and low- cost Remotely Piloted Aircraft System", 2019. https://doi.org/10.1016/j.jag.2019.101909

[5] J. Cuevas, A. Chua, E. Sybingco, and E. Bakar, "Identification of River Hydromorphological Features Using Histograms of Oriented Gradients Cascaded to the Viola-Jones Algorithm", International Journal of Mechanical Engineering and Robotics Research Vol. 8, No. 2, March 2019. https://doi.org/10.18178/ijmerr

[6] J. Cuevas, A. Chua, E. Sybingco, and E. Bakar, "Identification of river hydromorphological features using Viola-Jones Algorithm", 2016 IEEE Region 10 Conference (TENCON), 2016. https://doi.org/10.1109/TENCON.2016.7848439

[7] E. Belyaev and S. Forchhammer, "Drone HDR Infrared Video Coding via Aerial Map Prediction", 2018. https://doi.org/10.1109/ICIP.2018.8451554

[8] V. Delica, A. Orquia, J. Piquero, E. Reynaldo, J. Ilao, E. Sybingco, M. Roque, A. Chua, J. Katupitya, and H. Jayakody, "A New Sliding Mode Controller Implementation on an Autonomous Quadcopter System", International Journal of Automation and Smart Technology, vol. 9, no. 2, 2019. https://doi.org/10.5875/ausmt.v9i2.1876

[9] I. Gue and A. Chua, "Development of a Fuzzy GS-PID Controlled Quadrotor for Payload Drop Missions", Journal of Telecommunication, Electronic and Computer Engineering, vol. 10, Issue 1-5, pp. 55-58, 2018.

[10] C. Dim, F. Nabor, G. Santos, M. Schoeler, and A. Chua, "Novel Experiment Design for Unmanned Aerial Vehicle Controller Performance Testing", IOP Conference Series: Materials Science and Engineering, vol. 533, no. 1, 2019. https://doi.org/10.1088/1757-899X/533/1/012026

[11] A. Asares, P. Ko, J. Minlay, B. Sarmiento, and A. Chua, "Design of an Unmanned Aerial Vehicle Blimp for Indoor Applications", International Journal of Mechanical Engineering and Robotics Research, vol. 8, no. 1, 2019, Pages 157-161. https://doi.org/10.18178/ijmerr.8.1.157-161

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2016. https://doi.org/10.1109/TPAMI.2016.2577031

[13] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders, "Selective Search for Object Recognition", 2012. https://doi.org/10.1007/s11263-013-0620-5

[14] R. Girshick, "Fast R-CNN", 2015. https://doi.org/10.1109/ICCV.2015.169

[15] E. Barbe, Aerial view of palm trees on a caribbean island with a white sand beach and turquoise water- Sint-Vincent and the Grenadines, n.d. [Online]. Available: https://stock.adobe.com/ca/images/aerial-view-of-palm-trees-on-a-caribbean-island-with-a-white-sand-beach-and-turquoise-water-sint-vincent-and-the-grenadines/208681208. [Accessed: 19-Aug-2019]

[16] Altizure, Altizure android app quick start guide, 2016 [Online]. Available: https://www.youtube.com/watch?v=3lZ7DJYLxXc [Accessed: 19-Aug-2019]

[17] Drones Made Easy, Map Pilot for DJI, n.d. [Online]. Available: https://www.dronesmadeeasy.com/Articles.asp?ID=254 [Accessed: 19-Aug-2019]

[18] Foxtech, How to make mapping waypoints in Mission Planner for Nimbus Mapping VTOL, 2018 [Online]. Available: https://www.youtube.com/watch?v=eMJbNZennM8 [Accessed: 19-Aug-2019]

[19] C. Crespo, J. Paredes, J. Villamin, E. Sybingco, C. Llorente, A. Abad, M. Cabatuan, "Computer-based surveying system capable of human detection and head count with picture and location transfer from autonomous quadrotor", B.S. thesis, De La Salle University, Manila, Philippines, 2016.

Publisher: Chinese Institute of Automation Engineers (CIAE)
ISSN: 2223-9766 (Online)