Ghaf Tree Detection from Unmanned Aerial Vehicle Imagery Using Convolutional Neural Networks

1st Guoxu Wang
Department of Computer Science
Loughborough University
Loughborough, UK
G.Wang@lboro.ac.uk

2nd Andrew Leonce
College of Technological Innovation
Zayed University
United Arab Emirates
Andrew.Leonce@zu.ac.ae

3rd E.A. Edirisinghe
School of Computing & Mathematics
Keele University
Keele, UK
E.Edirisinghe@keele.ac.uk

4th Tamer Khafaga
King Salman Royal Nature Reserve
Kingdom of Saudi Arabia
t.khafaga@ksrnr.gov.sa

5th Gregory Simkins
Shaybah Wildlife Sanctuary
Kingdom of Saudi Arabia
gregory.simkins@aramco.com

6th Umar Yahya
Department of Computer Science
Islamic University
Kampala, Uganda
umar.yahya@iuiu.ac.ug

7th Moayyed Sher Shah
King Salman Royal Nature Reserve
Kingdom of Saudi Arabia
m.shah@ksrnr.gov.sa

2023 International Symposium on Networks, Computers and Communications (ISNCC) | 979-8-3503-3559-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISNCC58260.2023.10323713

***Abstract*—The Ghaf is a drought-resilient tree native to some parts of Asia and the Indian Subcontinent, including the United Arab Emirates (UAE). To the UAE, the Ghaf is a national tree, and it is regarded as a symbol of stability and peace due to its historical and cultural importance. Due to increased urbanization and infrastructure development in the UAE, the Ghaf is currently considered an endangered tree, requiring protection. Utilization of modern-day aerial surveillance technologies in combination with Artificial Intelligence (AI) can particularly be useful in keeping count of the Ghaf trees in a particular area, as well as continuously monitoring unauthorized use to feed animals and to monitor their health status, thereby aiding in their preservation. In this paper, we utilize one of the best Convolutional Neural Networks (CNN), YOLO-V5, based model to effectively detect Ghaf trees in images taken by cameras onboard light-weight, Unmanned Aircraft Vehicles (UAV), i.e. drones, in some areas of the UAE. We utilize a dataset of over 3200 drone captured images partitioned into data-subsets to be used for training (60%), validation (20%), and testing (20%). Four versions of YOLO-V5 CNN architecture are trained using the training data subset. The validation data subset was used to fine tune the trained models in order to realize the best Ghaf tree detection accuracy. The trained models are finally evaluated on the reserved test data subset not utilized during training. The object detection results of the Ghaf tree detection models obtained by the use of four different sub-versions of YOLO-V5 are compared quantitatively and qualitatively. YOLO-V5x model produced the highest average detection accuracy of 81.1%. In addition, YOLO-V5x can detect and locate Ghaf trees of different sizes moreover in complex natural environments and in areas with sparse distributions of Ghaf trees. The promising results presented in this work offer fundamental grounds for AI-driven UAV applications to be used for monitoring the Ghaf tree in real-time, and thus aiding in its preservation.**

***Index Terms*—Convolutional Neural Networks, Ghaf tree, Object detection, YOLO-V5, Drone imagery**

979-8-3503-3559-0/23/$31.00 ©2023 IEEE

I. Introduction

The Ghaf tree also known as the tree of life by local people in Bahrain and much of Arabia, is a drought-resilient tree capable of withstanding the extreme harsh conditions of a desert environment [1]. The Ghaf tree, scientifically known as Prosopis cineraria [2], can survive in extremely dry and hot weather for hundreds of years with no artificial irrigation required. In the United Arab Emirates (UAE) particularly, the Ghaf was declared a National tree in 2008 due to its historical and national importance [3], [4]. The leaves of Ghaf trees have historically been used as food for camels, while its tender leaves are still used in the UAE to make salads and for various medicinal purposes. Like any other natural entity in the environment, the Ghaf trees have, in the recent years, increasingly become threatened by the ever expanding human activity in the UAE as a result of urbanization and infrastructural development projects [3]. Given the arid environment in which the Ghaf trees exist, aerial surveillance systems such as Unmanned Aerial Vehicles (UAV) based imagery are naturally the preferred monitoring mechanism for aerial monitoring of habitats in such environments. Specifically, light-weight UAV-based (i.e., Drone) imagery and sensing has in the recent

past been utilized in detecting and mapping woody species' encroachment in subalpine grassland [5], estimating carbon stock for native desert shrubs [6], and several other desert and forest monitoring applications [7]–[10]. While UAV imagery has enabled large scale high resolution and fast landscape mapping, the use of this significant imagery data is still largely limited to offline use, with much more to be realized for real-time applications [9], [11], [12].

More recently, deep learning, a branch of Artificial Intelligence (AI), has become the mainstream technology for solving object detection and classification problems in computer vision [13], replacing the traditional machine learning based approaches. The reason for the superior performance of this new approach is that deep learning allows for the automation of feature engineering automation as opposed to classical machine learning approaches where feature engineering is not automated. Popular deep neural networks used in image processing applications, (i.e., applications on 2D data), Convolutional Neural Networks (CNNs), can be divided into two categories. The first category is the Region-based CNN (R-CNN) with popular variations being, Fast R-CNN, and Faster R-CNN [14]–[16]. They are basically two-stage networks that generate region proposals in an initial stage, and then do classification and regression on the region proposals. The other category is the one-stage CNN architectures such as You Only Look Once - unified, real time object detection (YOLO) and single shot detector (SSD) [17], [18]. These architectures basically use only one CNN network to directly predict the categories and locations of different targets.

Over the past 5 years, YOLO has evolved from YOLO-V1, V2, V3, V5, V6, V7 [19] and the latest being V8. According to our recent research [20] and other related work [21]–[24] YOLO-V5 performs better than the previous versions of YOLO (i.e. V1, V2 and V3), SSD and R-CNN architectures for most object detection and classification tasks. YOLO-V5 is the more established version of the original family of YOLO CNNs and has hence been used in the proposed research. However, it is noted that YOLO V6 [25], V7 [26] and V8 [27], have been designed to achieve faster convergence, marginally better object localisation and classification accuracy and faster deployment times.

In this paper we investigate the use of four YOLO-V5 sub-variants representing DNN architectures of different depth, S (Small-shallowest), M (Medium), L (Large) and X (Extra Large).

For clarity of presentation, this paper is divided into five sub-sections. Section-1 provided an introduction to the application and research context. Section-2 provides the methodology to be used including dataset preparation, data labelling and training the Deep Neural Network models, YOLO-V5, S, M, L and X. Section-3 presents the Ghaf Tree detection experimental results and a detailed analysis of the performance of the four models trained. Finally, Section-4 concludes with an insight to future work and suggestions for improvements of the established DNN models.

II. Methodology

The workflow of the proposed method is shown in Figure 1. It includes two main stages: training stage (including training and validation) and testing stage. The details of each phase (i.e. data preparation, training, validation and testing) are described in the following sections [28].

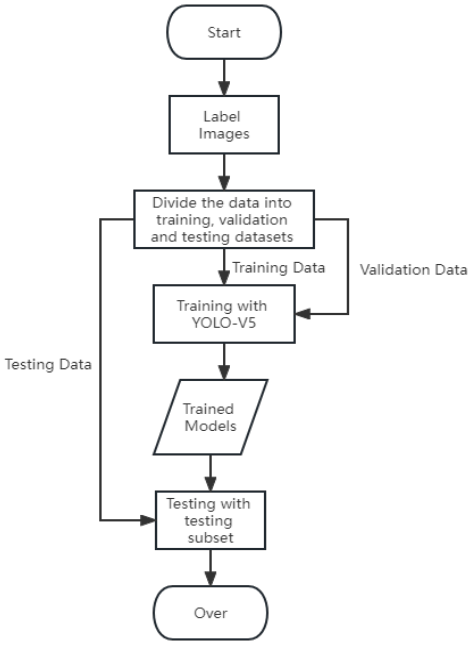


Fig. 1. The workflow of the proposed method

A. Data Preparation

A total 368 images containing 5000 Ghaf trees were randomly selected from a large number of images taken during a drone's flight. Drone imagery was collected by the Dubai Desert Conservation Reserve (DDCR) team with DJI Phantom-4 and DJI Mavic 2 Pro drones flying at different heights/altitudes. The selected image dataset was then divided into three data subsets for training (60%), validation (20%) and testing (20%). In the dataset used, as some of the images have a significantly larger number of Ghaf trees as compared to some others, the number of images in the training, validation and test data subsets were 298, 30 and 40, respectively. To start the training phase, each Ghaf tree in the training and validation dataset subsets was labeled with a bounding box using the labelImg tool and was labelled as type "Ghaf". It is specifically noted that some Ghaf trees can contain a number of canopies that grow from the same root structure, while some have only one canopy. Therefore it is often impossible to judge whether some adjacent canopies belong to the same root structure (as sand covers or occludes most of the roots and trunks) and hence form a single Ghaf tree. Therefore in this paper rather than attempting to detect a Ghaf tree, we attempt to detect Ghaf trees canopies. It should be therefore

noted that counting canopies, for example, will not allow us to count the total number of Ghaf trees.

TABLE I
NUMBER OF LABELED GHAF TREE CANOPIES IN EACH DATA SUBSET

Dataset	Number of labeled Ghaf tree canopies
Training	3200
Validation	900
Testing	900

The labeled data of the training data subset is used to train the four sub-versions of YOLO-V5 CNN. The training is for a single class of an object, 'Ghaf Tree' and hence a Ghaf tree is detected by differentiating it from its background. Similarly, the tagged Ghaf trees from the validation dataset subset is used to fine tune the training model of YOLO-V5 CNN when determining the optimal parameters of the model. Finally, the test dataset subset is used to evaluate the performance of the trained model. The labelled Ghaf trees in the validation set is used during training to optimize the network parameters, whilst the labelled Ghaf trees in the test dataset is used as benchmark data to determine the accuracy of prediction.

When labelling data for training and validation, when Ghaf trees are enclosed within rectangles, the rectangles may contain Ghaf trees of different sizes and may overlap or be obscured by other Ghaf trees or objects. Moreover, they may have different backgrounds (i.e. sand, bushes/shrub undergrowth, etc.). It is therefore important to capture rectangles of image pixels with the above possible variations for testing and training, as it will effectively test the generalizability of the trained CNN model for subsequent Ghaf tree detection tasks.

B. Network Architecture, Training and Detection

1) YOLO-Version 5 and Training: During training, four sub-versions of YOLO-V5 deep neural network, that includes YOLO-V5s (small), YOLO-V5m (medium), YOLO-V5l (large) and YOLO-V5x (extra-large), were trained by using the data related to the annotated rectangles containing Ghaf trees of the training data subset. Each sub-version of the YOLO-V5 network has a different model depth, but their designs are based on the same underlying structure, composed of three main parts: backbone, neck and head. The architectures of the S, M, L and X YOLO-V5 sub-versions indicate that the depth of the network increases [29] from S to X. Figure 2 [30] shows the network architecture of YOLO-V5s, which also represent the core basic architecture of all network sub-versions of YOLO-V5.

The model backbone of YOLO-V5 is used to extract basic features from a given input image. It is designed based on the Cross Stage Partial Network (CSPNet) [31] to extract advanced features while maintaining high accuracy and reducing model processing time. The model neck is mainly used to collect feature maps from different stages of the model trunk to generate feature pyramids. In YOLO-V5, the Path Aggregation Network (PANet) [32] is used to obtain the feature pyramid. The important function of a feature pyramid is to help the model identify the same objects with different sizes and scales. Finally, the model head is used for the final detection part of YOLO-V5. The design architecture of the model head of YOLO-V5, is the same as that of YOLO-V3 and YOLO-V4. It applies Anchor Boxes [33] to the final feature map and generates the final output vector with object score, class probability and boundary box coordinates.

As shown in Figure 2, YOLO-V5 has many sub-components in each part (i.e. backbone, neck and head) of its network, such as Focus, CBL, CSP and SPP modules. The Focus module is a module for processing input images. It uses four parallel slicing operations to create feature maps. The CBL module is a basic module. It uses a Convolution operation (Conv) combined with Batch Normalization (BN) [34] and leaky-ReLU [35] activation function to extract features. The CSP module is a module designed based on CSPNet. There are two types of CSP modules in YOLO-V5 network, i.e as CSP1 and CSP2. The CSP1 and CSP2 modules are applied to the backbone and neck of YOLO-V5 network. CSP1 contains one CBL module and N, Residual (RES) units. CSP2 contains N + 1 CBL modules. CSP1_UN and CSP2_N module runs under the same operation, divides the input feature mapping into two parts, and then integrates cross level features, where n represents the number of res units and CBL modules in CSP1 and CSP2 respectively. The more RES units and CBL modules, the deeper the network. Table 2 shows the usage of CSP modules in the four sub-versions of YOLO-V5 network. The SPP module is a module for mixing and collecting spatial features in the model backbone. It contains up to three Pool layers. The input feature is down sampled through three parallel Maximum Pool layers [36], and then the results are connected to the initial feature.

TABLE II
THE CSP MODULES USED IN DIFFERENT VERSIONS OF YOLO-V5 NETWORKS

Modules	YOLO-V5s	YOLO-V5m	YOLO-V5l	YOLO-V5x
CSP1	CSP1_1	CSP1_2	CSP1_3	CSP1_4
CSP1	CSP1_3	CSP1_6	CSP1_9	CSP1_12
CSP1	CSP1_3	CSP1_6	CSP1_9	CSP1_12
CSP2	CSP2_2	CSP2_4	CSP2_6	CSP2_8
CSP2	CSP2_2	CSP2_4	CSP2_6	CSP2_8
CSP2	CSP2_2	CSP2_4	CSP2_6	CSP2_8
CSP2	CSP2_2	CSP2_4	CSP2_6	CSP2_8
CSP2	CSP2_2	CSP2_4	CSP2_6	CSP2_8

All experiments were conducted on a PC equipped with an Intel Core i7-6850k CPU, NVIDIA GeForce gtx-1080ti GPU and 32 GB ram. The operating system on the computer used was, Windows10.

2) Ghaf Tree Detection: After completing the training process, the trained models of each of the YOLO-V5 network sub-version is used as a detector to detect the Ghaf trees in the test data subset. Each of the tests image of size 5472 $\times$3648 pixels is automatically adjusted to 608$\times$608 pixels by the network, while maintaining the original aspect ratio of

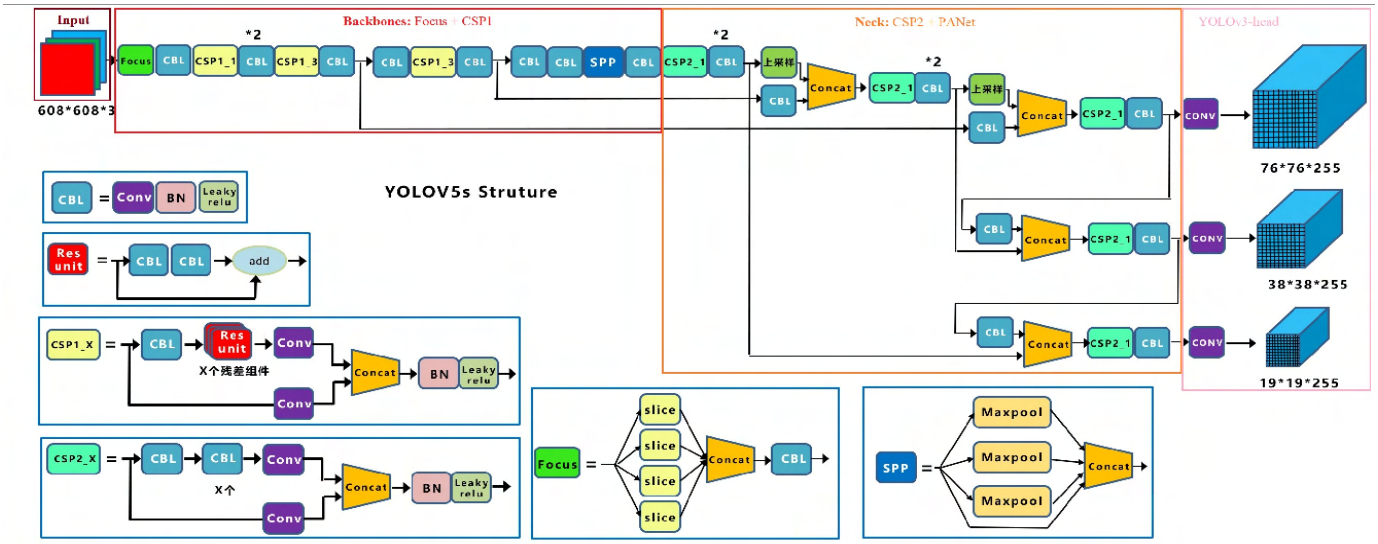


Fig. 2. The network architecture of YOLO-V5s

the input image. Then, the square image is processed through the network to extract features, and feature maps of three sizes, i.e., 76×76, 38×38 and 19×19 pixels are generated for prediction. The prediction under the feature map size of 76×76 pixels is used to detect small-size Ghaf trees, and the prediction under the feature map sizes of 38×38 and 19×19 pixels are used to detect medium-sized and large-size Ghaf trees, respectively. Under each prediction scale, each feature mapping unit predicts three bounding boxes using the three anchor box scales automatically learned in the training process. Each bounding box contains coordinates, width, height, a prediction confidence related score, and the class probability. The confidence score reflects the confidence level of the bounding box (i.e Ghaf tree) containing the object. If the confidence score (0-1 range) of the bounding box is low or zero, it means that the bounding box does not contain any objects. By setting a threshold, the model can delete bounding boxes with low confidence so that only bounding boxes containing objects of interest are retained.

III. Experimental Results and Analysis

In this section, we compare the performance of the Ghaf tree detection models generated from the four versions of YOLO-V5. Each model was trained with the same set of UAV images of the training data subset, validated on the same validation data subset and tested on the same test data subset. The performance of the four models are compared booth quantitatively and intuitively, below.

A. Quantitative Performance Comparison

A number of metrics can be used to analyze the performance of an object detector / model.

- TP (True Positive): The sample's true category is a positive example, and the model's anticipated result is also a positive example, indicating that the prediction is right.
- TN (True Negative): The sample's true category is a negative example, and the model predicts that it will be a negative example, which is right;
- FP (False Positive): The sample's true class is a negative example, but the model predicts a positive example, which is incorrect;
- FN (False Negative): The sample's true class is a positive example, but the model predicts a negative example, resulting in an incorrect prediction.
- IoU (Intersection over Union): IoU is a key notion in object detection. In general, it refers to the intersection of the bounding box and the model's projected Ground Truth. If the IoU is greater than an agreed threshold, we can conclude that the forecast was right.
- mAP (mean Average Precision): mAP can characterize the entire precision-recall curve (see Fig.3). The area under the precision-recall curve is mAP [In our experiments we set a threshold of 0.5].

Performance measures Accuracy, Recall and Precision can be derived as per the equations below, where TP, TN, FP, FN refers to the number of true positives, true negatives, false positives, and false negatives, respectively:

- Accuracy (all correct / all):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Recall (true positives / all actual positives):

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- Precision (true positives / predicted positives):

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

In determining the accuracy of an object detector, it is important to judge the accuracy based not only on the fact that an object has been correctly identified as being of a particular type, but also to determine how close the location of the object identified, is to the ground truth. Therefore, instead of using Accuracy, in this research we use mAP@0.5IoU as a measure to determine correctness of object detection. In our experiments we compared the performance of the four object detection models we obtained by training the four sub-versions of YOLO-V5, with models based on other popular Deep Neural Networks, SSD (Single Shot Multibox Detector) [37] and Faster R-CNN (Faster Region-based Convolutional Neural Networks) [38] (see Table-3).

The results tabulated in Table-3 also show that SSD requires significant amount of time for the convergence of training (i.e. completion of training) and also take significant amount of extra time for testing. Recall, precision and mAP@0.5IoU values also remain significantly lower than those of the YOLO-V5 models. Faster-RCNN took the lowest amount of time to complete training and has very good testing speeds, second only to YOLO-V5s, the shallowest YOLO-v5 sub-version. However, Accuracy, precision, mAP@0.5IoU values were much lower than in the case of the four YOLO-V5 models. Comparing the performance of the four YOLO-V5 models, it is observed that when the complexity/depth of the architecture increases, more time is taken for training and generally the same trend exists when it comes to testing, with YOLO-V5x taking significantly more time than sub-versions, m and l for testing. In comparison to other models, YOLO-V5x achieved the highest mean average precision (81.1%) in Ghaf tree detection, as shown in Table 3. Figure 3 illustrates the Precision vs Recall graph for YOLO-V5x indicating a mAP@0.5IoU value of 0.811.

TABLE III
PERFORMANCE COMPARISON OF DNN BASED OBJECT DETECTION MODELS

Model	Training Hours	Recall	Precision	mAP
SSD	**18**	**0.50**	**0.14**	**28.6%**
Faster-RCNN	**5**	**0.52**	**0.56**	**57.6%**
YOLO-V5s	**6**	**0.71**	**0.82**	**78.8%**
YOLO-V5m	**7**	**0.69**	**0.86**	**78.3%**
YOLO-V5l	**8**	**0.72**	**0.83**	**77.4%**
YOLO-V5x	**10**	**0.71**	**0.88**	**81.1%**

In summary, based on the objective performance values presented and discussed above, it can be concluded that the object detector models generated from YOLO-V5 and its sub versions are far superior in performance as compared to models generated by other popular CNNs such as SSD and Faster-RCNN. In particular the deeper the architecture, the objective performance improves when comparing the different sub-versions of YOLO-V5.

B. Subjective Performance Comparison

As shown by the objective performance results tabulated in Table 3, all four YOLO-V5 sub-versions can detect Ghaf trees

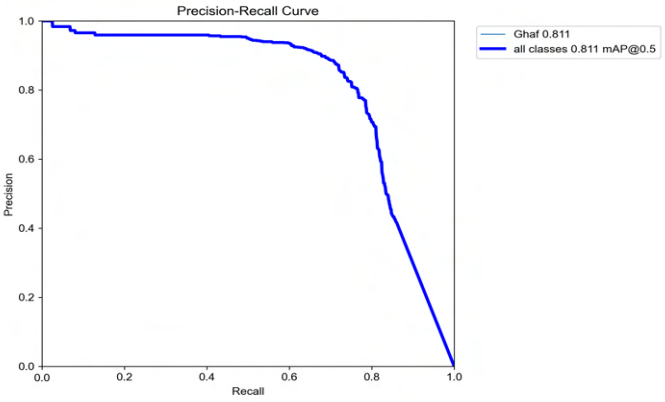
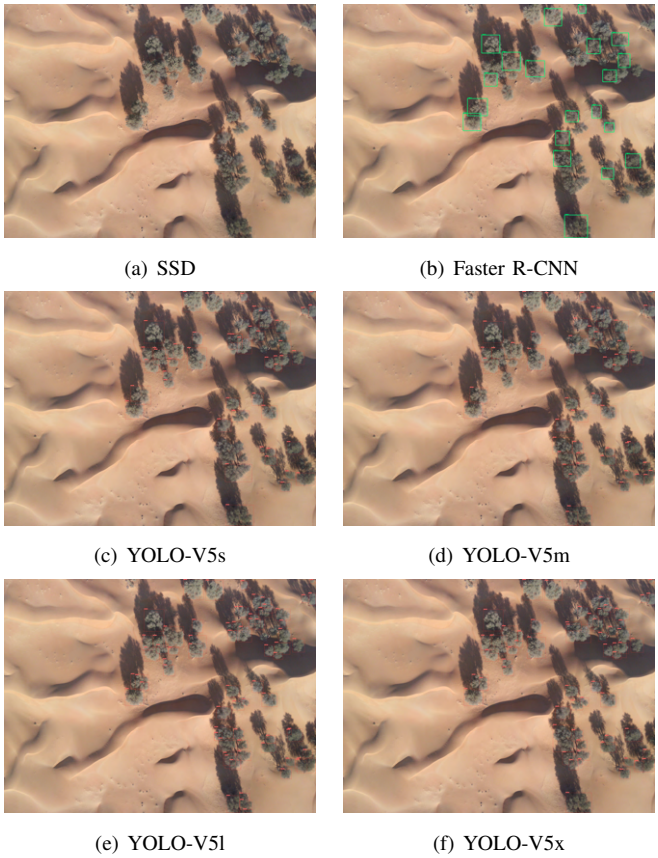


Fig. 3. Precision Vs Recall Graph



(a) SSD (b) Faster R-CNN

(c) YOLO-V5s (d) YOLO-V5m

(e) YOLO-V5l (f) YOLO-V5x

Fig. 4. The visual performance comparison of Ghaf tree detector models derived from DNN architectures, (a) SSD, (b) Faster-RCNN, (c) YOLO-V5s, (d) YOLO-V5m, (e) YOLO-V5l, and (f) YOLO-V5x

with a typically acceptable mAP of over 77.4%. To further analyze and compare the performance of the various models developed, this section provides a comprehensive subjective performance analysis. Figures 4-8 provide a number of different images taken from the drone footage of different areas that contain Ghaf trees. Some areas only contain Ghaf trees, and others contain other types of trees or plants.

Figure 4 illustrates the performance of the six models on

a desert area only containing Ghaf trees. Unfortunately, SSD based model did not pick up any of the Ghaf trees and The Faster-RCNN based model did not detect a number of Ghaf trees. The performance of the models created by the four sub-versions of YOLO-V5 were very much comparable. It is noted that in this image Ghaf trees have been captured at a high resolution with clear views of trees, with no other types of objects in the background.

Figure 5 illustrates the performance of models created by the four sub-versions of YOLO-V5, on a drone captured image of a higher altitude (hence trees appearing smaller) and in an area where there are other trees and objects. The yellow circles denote missed Ghaf trees. The model based on YOLO-V5x outperforms the models based on other YOLO-V5 sub-versions. YOLO-V5s misses some sparse canopies. YOLO-V5s, m and l misses Ghaf trees located at the boundary of the image.

Further inspecting the images included in Figure 6, it can observe that the models generated by training the four YOLO-V5 sub-versions, perform very well most of the time, and their operational / accuracy gaps only exists in some detail. YOLO-V5x performs marginally better when detecting small canopy, overlapped and close canopies.

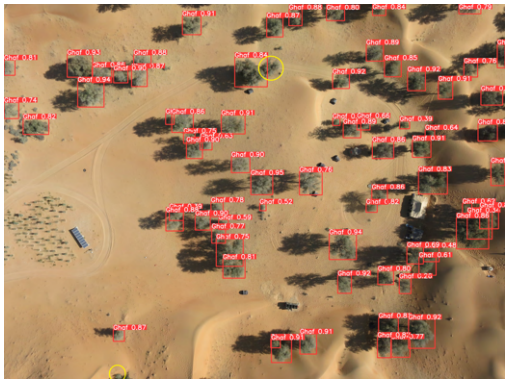
Figure 7 illustrates the use of the models created by the four sub-versions of YOLO-V5 in detecting Ghaf trees in a more complex area, that includes other trees. This image consists of Ghaf trees of a wider size variation. Comparing with the labelled ground truth image, the model generated by YOLO-V5x demonstrates a better performance as compared with the performance of the model created by YOLO-V5l. When the scene becomes complex, significantly more data is needed for training a deeper Neural Network. Thus, if we can have more high quality data for training YOLO-V5x, the results can still be improved.

Figure 8 illustrates the performance of the four models on another test image in which other different size of trees or plants exist in a complex area. In this specific case the model created by YOLO-V5m, performs better than other versions, rarely missing a Ghaf tree. Once again, the slightly less accurate detection capability of YOLO-V5l and YOLO-V5x can be attributed to the lack of substantial quantities of training data. The depth of the model and the amount of data will affect the actual detection situation. In certain situations, a certain model may perform better.

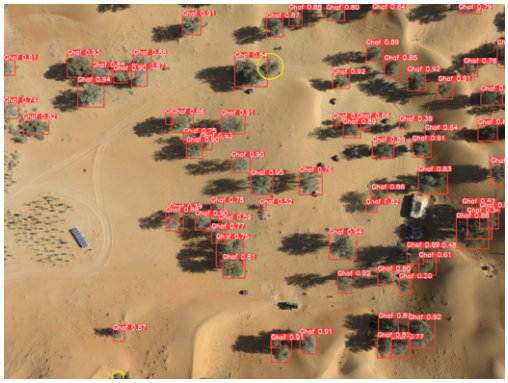
IV. Conclusion

In this paper we investigated the use of Convolutional Neural Networks in detecting Ghaf trees in videos captured by a drone, flying at different altitudes and in different environments that consists Ghaf trees. To the best of authors knowledge, this is the first attempt in using Convolutional Neural Networks in automatically detecting Ghaf trees, that poses a significant challenge to detecting them using traditional machine learning approaches. Despite the relatively small number of images utilized for training the DNNs in this work,

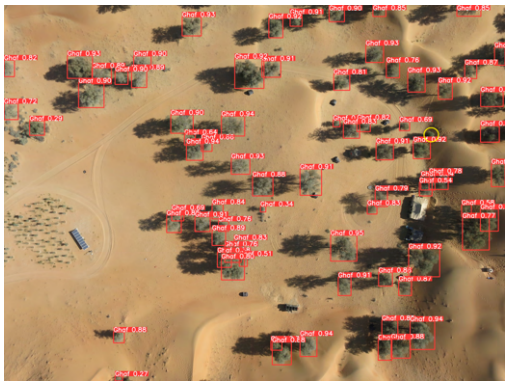
(a) YOLO-V5s



(b) YOLO-V5m

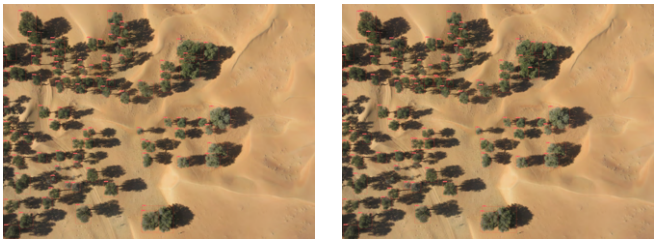


(c) YOLO-V5l

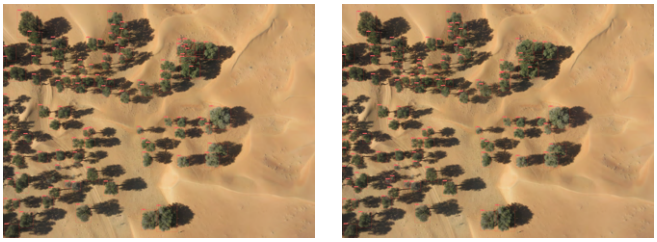


(d) YOLO-V5x

Fig. 5. The results of Ghaf tree detection in drone imagery using the YOLO-V5s, YOLO-V5m, YOLO-V5l, and YOLO-V5x

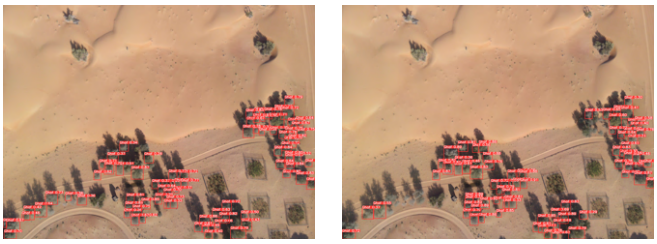


(a) YOLO-V5s (b) YOLO-V5m



(c) YOLO-V5l (d) YOLO-V5x

Fig. 6. The results of Ghaf tree detection in drone imagery using the YOLO-V5s, YOLO-V5m, YOLO-V5l, and YOLO-V5x

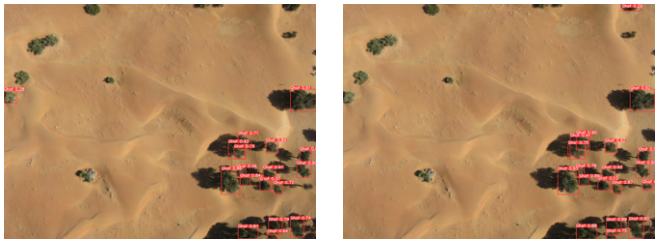


(a) YOLO-V5s (b) YOLO-V5m

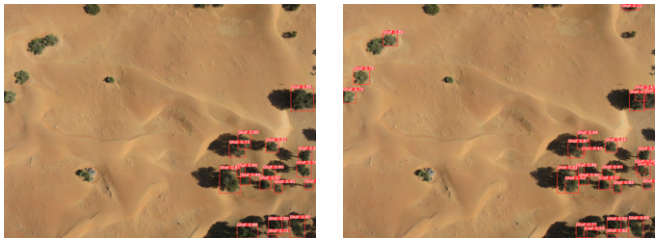
(c) YOLO-V5l (d) YOLO-V5x

Fig. 7. The results of Ghaf tree detection in drone imagery using the YOLO-V5s, YOLO-V5m, YOLO-V5l, and YOLO-V5x

the high mAP@0.5IoU value of 81.1% obtained by the YOLO-V5x based model in detecting Ghaf trees in approximately 78 MS, is a promising step towards achieving real-time detection using aerial imagery. The training time for model generation was high, approximately 10 hours and this was mainly due to hardware limitation of the computer utilized. The training time could be considerably reduced if a faster computer hardware was utilized. Models trained based on all other sub-versions of YOLO-V5 resulted in mAP@0.5IoU values of above 77.4%, whilst other popular DNNs such as SSD and Faster-RCNN performed less efficiently. Rigorous visual inspection of Ghaf tree detections obtained using all four sub-versions of YOLO-V5 revealed that YOLO-V5x particularly



(a) YOLO-V5s (b) YOLO-V5m



(c) YOLO-V5l (d) YOLO-V5x

Fig. 8. The results of Ghaf tree detection in drone imagery using the YOLO-V5s, YOLO-V5m, YOLO-V5l, and YOLO-V5x

outperforms the other YOLO-V5 networks at detecting Ghaf trees in scenarios where images are overlapping, blurred, obstructed with different backgrounds, and where there is a significant size variation of Ghaf trees.

This work utilized just over 5000 ghaf trees with 3200 of them used during training. If this number can be expanded to 10,000 + images, the detection performance will be further improved as the models would then be able to better generalize on new unseen data and accurately identify Ghaf trees. Dataset limitation notwithstanding, the results obtained in this work promise ground for real-time detection of the Ghaf using aerial surveillance, thus aiding in the efforts to preserve this endangered national tree of the UAE. The models can be used to design change detection software to identify damages to Ghaf trees based on drone captured aerial footage.

ACKNOWLEDGMENT

Authors acknowledge the support given by the Dubai Desert Conservation Reserve (DDCR) in capturing drone imagery and providing valuable advice in labelling the drone imagery utilized in this work. At the time of conducting this research the authors, Simkins, Khafaga and Shah, were employees of DDCR. The images used in this research were provided to Loughborough University by the DDCR, under a formal agreement.

REFERENCES

[1] R. Ahmad and S. Ismail, "Use of prosopis in arab/gulf states including possible cultivation with saline water in deserts," *Prosopis*, vol. 13, 1996.

[2] R. K. Kalarikkal, Y. Kim, and T. Ksiksi, "Incorporating satellite remote sensing for improving potential habitat simulation of prosopis cineraria (l.) druce in united arab emirates," *Global Ecology and Conservation*, vol. 37, p. e02167, 2022.

[3] D. Gallacher and J. Hill, "Status of prosopis cineraria (ghaf) tree clusters in the dubai desert conservation reserve," *Tribulus*, vol. 15, no. 2, 2005.

[4] V. BHARDWAJ, "Pods of prosopis cineraria (ghaf): A gift of nature for nutraceutical," *Journal of Global Ecology and Environment*, pp. 15–18, 2021.

[5] L. Oddi, E. Cremonese, L. Ascari, G. Filippa, M. Galvagno, D. Serafino, and U. M. d. Cella, "Using uav imagery to detect and map woody species encroachment in a subalpine grassland: advantages and limits," *Remote Sensing*, vol. 13, no. 7, p. 1239, 2021.

[6] M. M. Abdullah, Z. M. Al-Ali, and S. Srinivasan, "The use of uav-based remote sensing to estimate biomass and carbon stock for native desert shrubs," *MethodsX*, vol. 8, p. 101399, 2021.

[7] D. Gallacher, "Ecological monitoring of arid rangelands using micro-uavs (drones)," in *FINAL CONFERENCE PROCEEDINGS*, p. 181.

[8] R. Dainelli, P. Toscano, S. F. Di Gennaro, and A. Matese, "Recent advances in unmanned aerial vehicle forest remote sensing—a systematic review. part i: A general framework," *Forests*, vol. 12, no. 3, p. 327, 2021.

[9] M. Abdelkader, M. Shaqura, C. G. Claudel, and W. Gueaieb, "A uav based system for real time flash flood monitoring in desert environments using lagrangian microsensors," in *2013 International conference on unmanned aircraft systems (ICUAS)*. IEEE, 2013, pp. 25–34.

[10] M. M. Abdullah, Z. M. Al-Ali, M. T. Abdullah, and B. Al-Anzi, "The use of very-high-resolution aerial imagery to estimate the structure and distribution of the rhanterium epapposum community for long-term monitoring in desert ecosystems," *Plants*, vol. 10, no. 5, p. 977, 2021.

[11] A. C. Hill and Y. M. Rowan, "The black desert drone survey: New perspectives on an ancient landscape," *Remote Sensing*, vol. 14, no. 3, p. 702, 2022.

[12] L. Hashemi-Beni, J. Jones, G. Thompson, C. Johnson, and A. Gebrehiwot, "Challenges and opportunities for uav-based digital elevation model generation for flood-risk management: a case of princeville, north carolina," *Sensors*, vol. 18, no. 11, p. 3843, 2018.

[13] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 91–124, 2021.

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[19] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[20] T. Jintasuttisak, E. Edirisinghe, and A. Elbattay, "Deep neural network based date palm tree detection in drone imagery," *Computers and Electronics in Agriculture*, vol. 192, p. 106560, 2022.

[21] S. Bilik, L. Kratochvila, A. Ligocki, O. Bostik, T. Zemcik, M. Hybl, K. Horak, and L. Zalud, "Visual diagnosis of the varroa destructor parasitic mite in honeybees using object detector techniques," *Sensors*, vol. 21, no. 8, p. 2764, 2021.

[22] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using yolo-v5 model." *BioResources*, vol. 16, no. 3, 2021.

[23] L. Wang and W. Q. Yan, "Tree leaves detection based on deep learning," in *Geometry and Vision: First International Symposium, ISGV 2021, Auckland, New Zealand, January 28-29, 2021, Revised Selected Papers 1*. Springer, 2021, pp. 26–38.

[24] J. Zhao, X. Zhang, J. Yan, X. Qiu, X. Yao, Y. Tian, Y. Zhu, and W. Cao, "A wheat spike detection method in uav images based on improved yolov5," *Remote Sensing*, vol. 13, no. 16, p. 3095, 2021.

[25] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[26] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[27] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, "Yolov6 v3. 0: A full-scale reloading," *arXiv preprint arXiv:2301.05586*, 2023.

[28] B. Alvey, D. T. Anderson, A. Buck, M. Deardorff, G. Scott, and J. M. Keller, "Simulated photorealistic deep learning framework and workflows to accelerate computer vision and unmanned aerial vehicle research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3889–3898.

[29] D. Thuan, "Evolution of yolo algorithm and yolov5: The state-of-the-art object detention algorithm," 2021.

[30] D. Jiang, "A complete explanation of the core basic knowledge of yolov5 in the yolo series," Zhihu, 01 April 2022. [Online]. Available: https://zhuanlan.zhihu.com/p/172121380. [Accessed 10 May 2022].

[31] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[32] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[33] Y. Zhong, J. Wang, J. Peng, and L. Zhang, "Anchor box optimization for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1286–1294.

[34] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *Advances in neural information processing systems*, vol. 31, 2018.

[35] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky relu," in *2020 IEEE Symposium on Computers and communications (ISCC)*. IEEE, 2020, pp. 1–7.

[36] N. Murray and F. Perronnin, "Generalized max pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2473–2480.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.