Identifying the Maturity Level of Coconuts using Deep Learning Algorithms

Harsh Mittal[1[0000-0002-9955-1405]], Nuthalapati Hemalatha[2[0000-0003-0728-7924]], and Ravi Kumar Mandava[3*[0000-0001-6186-9138]],

[1]Department of Engineering Physics, Delhi Technological University, India
[2]Department of Electronics and Communication Engineering, R.V.R & J.C College of Engineering, Guntur, India
[3]Maulana Azad National Institute of Technology, Bhopal MP 462003, India
*ravikumar1013@gmail.com

Abstract. Identifying the maturity level of the coconuts is challenging for coconut cultivators. Up to now, many coconut cultivators in India are manually determining the maturity level. It is difficult for a person can climb the coconut tree and identify the maturity level of the coconuts. Few farmers started the deployment of robots to harvest the mature coconuts from the tree. In this process, the robot determines the maturity of coconuts using image processing techniques. This article discusses an efficient and accurate method for determining the maturity level of coconuts. The authors proposed a dataset for detecting raw and ripened coconuts in this work. Further, to train the dataset, the authors used various deep learning algorithms such as YOLO-V5s, YOLO-V4Tiny, and Mobilenet-SSD, respectively. Finally, the results related to different deep learning algorithms have been compared in terms of Precision, Recall, and F1 score.

Keywords: Coconuts, maturity level, image processing, dataset, and deep learning.

1 Introduction

Day by day the importance of computer vision is increasing in all agricultural applications. Instead of using the human eye, the advanced technology that is, machine vision which employs a camera to recognise the object using image processing technique [1]. The role of image processing has become extensively used in coconut harvesting applications and is has played a crucial role in its growth [2]. India is also one of the country to produce the coconuts in high capacity. Due to the height of the tree and uneven surface of the trunk, many of coconut climbers are not interested for climbing the trees. Few researchers, had developed various climbing mechanisms to climb the coconut trees. It has been found that the developed climbing mechanisms requires human effort. The diameter of the tree trunk can vary, and the surface of the tree may damage making it challenging to employ tree-climbing equipment [3].

The aim of this research focuses on the artificial intelligence (AI)-based design of that intelligent system. The developed intelligent system predicts a ripe coconut in the current image of the coconut tree's crown using the prior information. The proposed dataset is designed to classify coconuts into two categories—raw coconut and matured/ripened coconut—based on their size, shape, colour, and other visual characteristics. This dataset has been trained on several cutting-edge computer vision models and deployed to a coconut harvesting robot that autonomously recognises the coconuts, harvesting and collecting from the tree. One of the key function that must be carried out in real-time for the automated coconut harvesting machine is the identification of mature coconuts in the coconut tree crown. Further, an intelligent system can be connected to the machine to check the performance. Using a computer board and Jetson Nano board, analyses the image with a dataset and recognises the different stages of the coconut, it is possible to identify the coconut in the current capture image [4].

The study also examines cutting-edge machine learning techniques for plant disease identification using various data sources [5]. The key data collecting modalities the internet of things, unmanned aerial vehicle imaging, satellite imaging and ground imaging are combined with conventional and deep learning techniques. Additionally, this study looks at how data fusion is used in current studies to diagnose diseases. It demonstrates the benefit of using intelligent data fusion approaches to combine data from many sources to enhance the prediction of plant health status and outlines the key difficultiesthis sector now faces.

Many researchers have been developed several deep learning algorithms for better deployability, lower latency, higher object detection and classification accuracy. One such algorithm is the YOLO [6], in which authors frame the detection of object detection as a regression problem. From the complete images the bounding boxes and class probabilities are directly predicted by a single neural network. Therefore, detection of pipeline consists of a single network, and it canbe tuned from beginning to end. In real time the YOLO model processes 45 frames per second. Further, the a scaled-down version of the YOLO network processes 155 frames per second to perform the real-time detectors in a map by a factor of two. Modern detection methods make fewer predictions of false positives in the background, while YOLO causes more localization errors. Last but not least, YOLO picks up very broad representations of objects. Extrapolating from natural images to other domains like artwork performs better than other detection techniques, including R-CNN [7].

Another cutting-edge algorithm for routine detection tasks like PASCAL VOC [8] and COCO [9] is the YOLO-V2 [10]. Later on, YOLOv2 algorithm can run at different scales using an innovative, multi-scale training strategy, providing a simple tradeoff between speed and accuracy. On VOC 2007, YOLOv2 received 76.8 from map at 67 FPS. Further, at 40 frames per second, YOLOv2 achieves 78.6 map, which exceeding cutting-edge techniques like Faster RCNN [11] with ResNet [12] and SSD [13] while still operating a great deal quicker. Finally, the authors provide a technique for training object detection and classification. The above technique is used to train the ImageNet classification dataset and the COCO detection dataset [14] using YOLO9000 [15].

2 Methodology

The object of this research is to develop a custom dataset for classifying coconuts into raw and ripened or matured, based on the visual data only, i.e. video stream through the pi-camera. Thus, in our analysis, we sunder our experimentation into the subsequent parts: Dataset collection, Object detection and Classification.

2.1 Data Collection

A dataset has been prepared to identify the raw coconuts and mature coconuts based on their maturity levels. The raw coconuts are labelled as raw coconut, while the mature coconut has been labelled as matured coconut. The dataset contains 1021 images, out of which a total number of 900 images required for training the models, while 71 images required for validation and 50 for the testing set. A 88% and 7% of the data were used for the train-validation split. Further, the images were stretched and resized to 416 x 416. and they applied various augmentation techniques like output per training example, horizontal flips, rotation of images between -15◦ and 15◦, application of noise per pixel and blurring upto 10 pixels etc. to make the dataset more robust and sturdy. The above dataset was developed with the help of Roboflow. Figure 1 shows the training image of ripened coconut.

Fig. 1. Ripe Coconut image for Training.

2.2 Object Detection and Classification

In the present research problem, various deep learning algorithms such as, YOLO-V5, YOLO-V4 Tiny and Mobilenet-SSD are used to train the datasets.

YOLO-V5: YOLO-V5 is not considered as an official version of the YOLO family of models. The major concern of YOLO-V5 remains the accuracy of the model and slower inference compared to various models available, the precision of the model and the faster training make it a compelling choice and worth the trade-offs.

YOLO-V5 is a single-stage object detector, it also has three important steps, which are:

a) Model back bone
b) Model neck
c) Model head

Model backbone is mainly useful for extracting the features from the given input images. YOLO-V5 uses Cross Stage Partial Networks (CSP), as its backbone for extracting features from the given input images. Further, the model neck has the main objective to generate the feature pyramids. This will help to generalize the model well on object scaling, i.e. helping the model to recognize the objects on different scales and it also helps the model to enhance performance on unseen data. YOLO-V5 uses Path Aggregation Network for Instance Segmentation (PANet). Later on, the model head is used to detect the final part, which applies anchor boxes on features and creates final output vectors, with classes, probabilities, bounding boxes etc. The algorithm YOLO-V5 is used in YOLO-V4 and V3 versions. The YOLO-V5 uses Leaky Relu in the hidden layers, and the sigmoid activation function in the final output layer. The original architecture of the YOLO-V5 uses SDG optimization function and the Binary Cross-Entropy with Logits Loss as the Loss Function. The algorithm achieves a Precision of 0.8940, an F1 Score of 0.87, and Recall of 0.8472.

YOLO-V4 Tiny: YOLO V4, is one of the finest computer vision algorithms, with higher accuracy, lower latency and higher inference rates, which it utilizes due to the optimizations in parallel computing, and thus makes it one of the most popular choices for deployment. Many other features have been proposed to increase the accuracy of conventional neural networks (CNNs). It is necessary to evaluate these feature combinations in practice on sizable datasets and to theoretically support the outcome. While some aspects, like the residual connections and batch-normalization are relevant to the majority of datasets, models, and tasks, others only function on specific models, for specific issues, for small scale datasets only. Cross Stage Partial-connections (CSP), Weighted Residual Connections (WRC), Cross mini Batch Normalization (CmBN), Self-adversarial training (SAT), and Mish-activation are among the universal features that the authors believe to exist. Later on, the authors integrate some of the new variation of features such as, CSP, CmBN, SAT, WRC, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss to provide cutting edge outcomes: 46.5% AP (65.7% AP50) for the MS COCO dataset at a real-time frame rate of about 65 FPS on the Tesla V100. YOLO-V4 can be broken down into the following major components:

a) Backbone Network - Feature Formation
b) Neck - Feature Aggregation
c) Head - Final Detection

Usually, ImageNet Classification is used as pre-training for the model's backbone network.

The backbones CSPResNext50, CSPDarknet53, and EfficientNet-B3 were taken into consideration. DenseNet creates as the basis for both the CSPRes- Next50 and the CSPDarknet53. Further, DenseNet was created to connect the various layers of convo-

lutional neural networks of the various goals such as, to enhance the feature propagation, encourage the network to reuse features, and decrease the number of network parameters. Further, the DenseNet has been improved as CSPDarknet53 and CSPResNext50. Therefore, the feature mAP of the base layer is distinct by copying it, transfer one copy through the dense block, and transfer to the next stage. By transmitting an unedited version of the feature mAP, the CSPDarknet53 and CSPResNext50 aims to diminish the processing bottlenecks in the DenseNet and enhance learning. The backbone network of the final YOLOv4 network uses CSPDarknet53.

For the network's feature aggregation, YOLOv4 employs PANet. The fact that NAS-FPN and BiFPN were created concurrently, even though the authors don't explain their reasoning, suggests that they might also be a good option. To broaden the receptive area and detached the most crucial characteristics from the backbone, YOLOv4 adds an SPP block following CSP- Darknet53. With anchor-based detection stages and three levels of detection granularity, YOLOv4 deploys the same YOLO head as YOLOv3 for detection, which are:

a) Bag of Freebies
b) Self-Adversarial Training (SAT)
c) Bag of Specials

The authors employ DIoU NMS to distinguish anticipated bounding boxesfor the activation function. It would be helpful to quickly select the optimal bounding box because the network might predict several bounding boxes for a single object. Cross mini-Batch Normalization was used to perform the batch normalization (CmBN). Drop Block regularization is a method that YOLOv4 employs to compel the network to acquire features that it might not otherwise relyupon.

Contrarily, YOLO-V4 Tiny minimizes computation cost by using two ResBlock-D modules in a ResNet-D network instead of two CSPBlock modules in Yolov4-Tiny. In order to reduce detection error, it also develops an auxiliary residual network block to extract more object feature information. Two consecutive 3x3 convolutions are utilized to create 5x5 receptive fields in the architecture of the auxiliary network to extract global features. Channel attention and spatial attention are also used to extract more useful information. Finally, it combines the backbone and auxiliary networks to create the upgraded YOLOv4-Tiny's entire network structure. The model achieves a precision of **0.8220**, Recall of **0.7028** and an F1 score of **0.7578**.

Mobilenet-SSD: The Mobilenet-SSD since its release, has reached new performance records, with a claimed 74% mAP, at 59 fps, on standard datasetssuch as PascalVoc and COCO. The SSD Mobilenet is based out of the SSD but for deployment onto mobile and embedded systems based applications. In SSD, the bounding boxes are generated at each resolution while lowering the resolution with Extra Feature Layers, following the extraction of the features using an arbitrary backbone. After calculating a total of 3000 bounding boxes usingthe output from the six levels of resolution, MobilenetSSD will filter out the bounding boxes using non-maximum suppression (nms). After each layer, SSDhas a total of 8732 bounding boxes which is more than that of YOLO. The mobilenetSSD relies on two loss functions:

a) Localization Loss
b) Confidence Loss

The localization loss is the difference between the expected boundary box and the ground truth box. Only accurate forecasts are penalized by SSD. The forecasts resulting from the successful matches should get more accurate. Un- favorable matches can be disregarded. The inability to make a class forecast constitutes the confidence loss. We penalize the loss for every accurate match prediction in accordance with the class's confidence score. When we make negative match predictions, we punish the loss based on the class "0" confidence score, which denotes that no item was detected. The model achieves a Precision of **0.7870**, Recall of **0.5414** and F1 Score of **0.6415**.

3 Results & Discussion

The results clearly indicate that Yolo-V5-s has an upper hand in metrics, while the easiest to deploy on edge devices like Raspberry Pi becomes the SSD-Mobilenet. While inference rates are best observed using Yolo-V4 Tiny because of the Darknet. All the models were trained on Tesla T4 GPUs,and deployed on Raspberry Pi 4, without using hardware accelerators like GoogleChoral. The results of this research are clearly indicative of the better inference ofYolo-V4, which makes it a practical choice. While using SSD, Mobilenet for rapidprototyping is suggested as vast documentation for deployment on Raspberry Pi.A comparison of the above mentioned algorithms can be referred from figure 2.

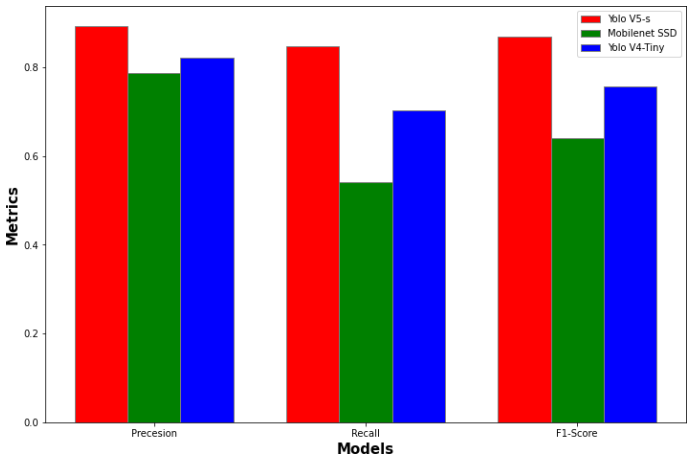

Fig. 2. Comparison of different object detection models.

4 Conclusions and Future Scope

Through this research, the authors proposed a dataset for detecting and classifying coconuts as raw and ripened based on visual input. The current research work rwas limited to the use of 1001 images of raw and matured coconut only, which makes the coconut dataset very small, and thus yields poorer results even on state-of-the-art models like the YOLO-V4. Also, very few augmentation techniques are applied, and the blurring per pixel has been kept at only up to 5%, which can be increased further for a better and more robust model. The other major concern with the proposed dataset, coconut, is that it has only two classes, but some acoustic techniques propose up to three different classifications of the coconut, thus their lies huge scope of improvement in the coconut dataset, with the increase in a number of images, classes and augmentations, which can either be done using manual techniques of labelling, as done in this research or using Generative Adversarial Neural Networks (GANs), which would help in creating synthetic data, as well as adding noise to the dataset, which would help in making the data more robust for real-time applications.

Acknowledgement: The authors are thankful to DST Govt. of India under SYST Scheme for funding the project number SP/YO/2019/1052(G).

References

1. Forsyth, D. and Ponce, J., 2011. Computer vision: A modern approach (p. 792). Prentice hall.
2. Patr´ıcio, D.I. and Rieder, R., 2018. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and electronicsin agriculture, 153, pp.69-81.
3. Megalingam, R.K., Sakthiprasad, K.M., Sreekanth, M.M. and Vivek, G.V., 2017, August. A survey on robotic coconut tree climbers–existing methods and techniques.In IOP Conference Series: Materials Science and Engineering (Vol. 225, No. 1, p. 012201). IOP Publishing.
4. Subramanian, P. and Sankar, T.S., 2021. Development of a novel coconut-tree-climbing machine for harvesting. Mechanics Based Design of Structures and Machines, pp.1-19.
5. Yang, X. and Guo, T., 2017. Machine learning in plant disease research. March, 31, p.1
6. Liu, C., Tao, Y., Liang, J., Li, K. and Chen, Y., 2018, December. Object detectionbased on YOLO network. In 2018 IEEE 4th Information Technology and Mechatron- ics Engineering Conference (ITOEC) (pp. 799-803). IEEE.
7. Li, J., Wong, H.C., Lo, S.L. and Xin, Y., 2018. Multiple object detection by a deformable part-based model and an R-CNN. IEEE Signal Processing Letters, 25(2), pp.288-292.
8. Hoiem, D., Divvala, S.K. and Hays, J.H., 2009. Pascal VOC 2008 challenge. WorldLiterature Today, 24.
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár,

P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

10. Jiang, P., Ergu, D., Liu, F., Cai, Y. and Ma, B., 2022. A Review of Yolo algorithm developments. Procedia Computer Science, 199, pp.1066-1073. Vancouver
11. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information pro-cessing systems, 28.
12. Haque, M.F., Lim, H.Y. and Kang, D.S., 2019, January. Object detection based on VGG with ResNet network. In 2019 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-3). IEEE.
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In European conference on com- puter vision (pp. 21-37). Springer, Cham.
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scalevisual recognition challenge. International journal of computer vision, 115(3), pp.211-252.
15. Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).