International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-3, January 2020

Optimal Deep Learning Model to Identify the Development of Pomegranate Fruit in Farms

Sripad Joshi, Sandeep Kumar Panda, Sathya A. R.

Abstract: In this study, estimating the maturing condition in gardens helps to enhance the process of post-harvesting. Collecting fruits on the basis of their developmental stage will minimize storage costs and maximize market value. Additionally, estimated ripeness of the fruits can be more useful for indicators for detecting water shortage and to determine the water used during irrigation. The purpose of the study is to develop the new direction of technology to detect the ripeness stage between two classes: ripe and unripe. We employ deep Neural Network (DNN) classifiers for the prediction of ripe and unripe class. The results of our proposed classifiers give the sensitivity 96.2%, specificity 94.2% with accuracy of results 94.5%, over a dataset of 200 images of each class. The ROC (receiver operating characteristic) area values curve close to 0.98 in all-class during training. We believe this is a notable performance that allows a suitable non-intrusive maturing prediction that will enhance cultivation techniques.

Keywords: Deep neural network, ripeness estimation, pomegranate segmentation.

I. INTRODUCTION

Automation in agriculture is thriving today globally, credit to the advancement in the Internet of Things (IoT) and deep learning approaches. The sophistication of deep learning approaches has made it possible for developing solutions to difficult real-life problems. Regarding agriculture, several problems such as plant disease identification from leaf images, appropriate crop cultivation based on several factors like weather, soil-type has been developed successfully. IoT is implemented to automate agricultural tasks such as monitoring humidity of soil, watering crops, etc. Although, only insubstantial work is done in the area of real-time fruit detection coupled with classifying them as ripe or unripe.

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author

Sripad Joshi, Department of Computer Science and Engineering, Faculty of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India. Email: varaprasad.fst@ifheindia.org

Sandeep Kumar Panda, Department of Computer Science and Engineering, Faculty of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India..Email: skpanda00007@gmail.com

Sathya AR*, Department of Computer Science and Engineering, **Faculty** of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India. Email: sathya.renu@gmail.com

The meticulous realization of the location of discrete fruits along with the information about their ripening condition contributes greatly to estimating the yield and time remained for harvest respectively. Automated robotic harvesting system can make use of this localization system [2,17]. The previous approaches involved considered detecting each ripening stage of the fruit as a YOLO object detection problem [4], which significantly increased the training costs. Hence, in our approach, we break down this problem into two parts. Our approach treats fruit detection as a YOLO object detection problem and ripening state detection as an image classification problem. This perspective benefits with having the flexibility of appending different classification tasks like disease detection of on-the-tree fruits and much more without vowing to much of the training cost. We, in this paper implement state-of-the-art object detection technology YOLO-v3, to detect and localize pomegranates in the orchards. We then send the detected pomegranate region images into the classifier which classifies them as ripe or unripe. The following comes under the scope of this paper:

- Image data accumulation.
- Annotation of the image data using YOLO mark, a python-based image annotation tool by A.B. Alexey [3]
- Training YOLOv3 on the data with darknet53 weights as backbone.
- Training the image classifier for detecting the ripening stages of the pomegranate fruit.

The structure of the paper is organized as follows

In section 2, we describe data accumulation, annotation tool, and its usage. Literature survey is discussed in section 3. Section 4 presents the proposed approach. In section 5 the experiment and results are detailed. Finally, section 6 concludes with future work directions and limitations.

A. Data Accumulation And Annotation

We primarily used image-net [14] open-source data in addition to google images for pomegranate images. Google's advanced search was used for the refined procurement of pomegranate fruit images from orchards. Total images were aggregated to 780. Below are five sample images from the data set.

Fig. 1. Images from the dataset in image annotation process

The images were annotated using the YOLO mark, a python based open-source image annotation tool [3].
The tool annotates images in a format suitable for YOLO. The following Fig. 2 shows the annotated samples of the pomegranate images which forms the dataset **D1**.

Fig. 2. Annotated images using YOLO mark tool -D1 dataset

Further, to build the classifier we segregated images of pomegranate as ripe and unripe which forms the dataset **D2,** the sample size in both case was 200 images. Both ripe and unripe images are presented in Fig 3(a&b), as ripe and Fig 3(c&d) as unripe.

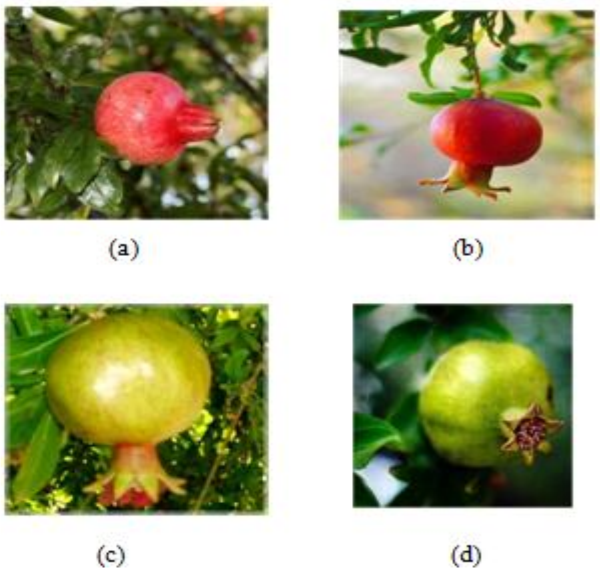

Fig. 3. (a)(b)Ripe Pomegranate, (c)(d) Unripe pomegranate - D2

To avoid over fitting of the classification model because of the small training data set, we apply image augmentation for generating more training data from the available image data. This helps in building a more robust model that can generalize better on new unseen images. We employed different image transformation methods such as flip, rotate, warp, zoom and lighting. Fig. 4 below shows an example image from the data set to which transformations were applied. Our model created eight artificial randomly transformed images for one actual image.

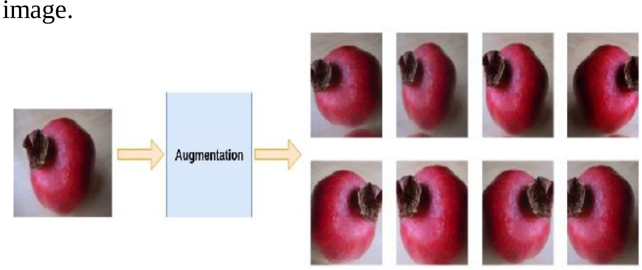

Fig. 4. Augmentation images of pomegranate fruits

II. RELATED WORK

Fruit detection in orchards has been addressed by Suchet Bargoti et al. [1]. In their study, they implemented Faster-RCNN based detection system to localize fruits like mangoes, almonds, and apples. However, they did not address the ripening stage detection problem. Another approach for fruit detection in the orchard was by A. Koirala [10]. They developed a new architecture based on YOLO-V3 and YOLO-V2 to detect mangoes in the orchards. Their architecture MangoYolo achieved an average precision of 0.983.D.Stajnkoa, M.Lakotaa, M.Hočevarb [12] used thermal cameras to capture apple images and applied different image processing algorithms to estimate the yield and size of apples in the orchards.

Ripening stage detection of fruits in orchards was addressed by Yunong Tian et al. In their paper[11] using an improved YOLO-V3 model they detected various developmental stages of pomegranate in orchards. Although they addressed the ripening detection problem, they trained different stages of apple growth as discrete objects and detected them directly using YOLO-v3. This approach would induce immense training costs if we required to add an additional detection stage to the model.Sunmok Kim et al. [13]. In their paper, a novice approach by Region of Interest (ROI) of input image segmentation using Yolo object detection is implemented. They concatenated frames of sign language into a broader image which is then sent as input to the object detection algorithm YOLO, which detected the hand region from the wide image, which is crucial for sign language learning. Convolution neural networks were employed for sign language learning.

Few researchers examined the improvements in the chemical and physical characteristics of various fruits during ripening[15, 16].

III. PROPOSED METHOD

For our purpose of pomegranate detection in orchards, we implemented YOLOv3, a darknet based object detection network architecture. In addition, we train a Resnet50 based image classifier for pomegranate ripening stage classification.

A. YOLOv3

YOLO-v3 [5], object detection algorithm by (Redmon and Farhadi, 2018) is an improvement over YOLO-v2 and YOLO [6]. Unlike proposal region-based approach algorithms such as state-of-the-art Faster RCNN, YOLO considers identification as a question of regression and generates boundary boxes and probability of classes by regression. This helps YOLO detect objects at an expeditious rate and work well in real-world scenarios.

YOLO partitions the representation of the input into a grid (S x S). If the center of an object to be expected falls within a certain grid, the grid is responsible for the object being observed. The grid predicts confidence scores of the objects present in it. If there is no object present in the grid, then the confidence score should be zero.

The individual grids also predict bounding boxes and conditional probabilities for the class.

YOLO uses Non-max suppression to choose the best bounding box if more than one bounding box detects the same object.

B. Image Classifier

Image classification task has become more robust, credit to the emergence of transfer learning. Pre-trained models such as VGG[9], Resnet [7] etc. help compensate when training data is sub-par. They provide a starting point for the training. We used Resnet architecture as the backbone for our image classifier. Resnet is faster to train, reduces the effect of vanishing gradient and is very accurate. We implemented Resnet50 architecture for our classification task.

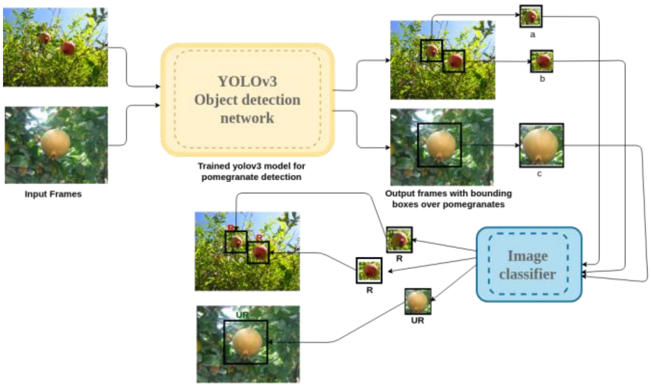

Fig. 5. (a,b,c) Pomegranate detected regions by YOLOV3 R: Ripe UR: Unripe

C. Our approach

We introduce a unique pipeline FruitYOLO where in we breakdown the real-time fruit ripening detection problem into two tasks.

Firstly, we train the YOLO-v3 detection network. For this purpose, we load the darknet.conv.53 pretrained weights as the backbone and proceed to fine tune the model with the annotated pomegranate dataset D1. Secondly, we build an image classifier for classification of the pomegranate images as ripe and unripe. For the classifier, we use the Resnet50 model as the backbone and fine tune it on the the D2 dataset.

In the real-time scenario, when a frame is sent to the YOLO-v3 network, it detects the pomegranate from the frame and sends those detected pomegranate regions to the trained classifier (as shown in Fig. 5). The classifier then classifies the fruit as ripe or unripe and sends back the label.

IV. RESULTS AND DISCUSSION

To train YOLOv3 the following are the initialization parameters.

Table-I: Parameters for Experimental analysis

Image Size	Batch size	Decay	Momentum	Training steps
608 x 608	64	0.9	0.00005	5000

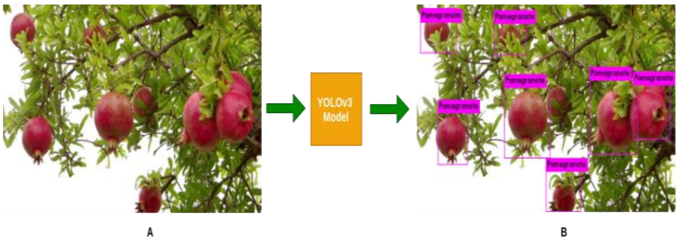

Fig. 6. A. Input frame to the Yolo model with output detections

We trained our YOLOv3 model on a Tesla k80 GPU for 5000 training steps. After training we achieved a mean accuracy precision of 75%.

In Fig. 6 above shows the output of the YOLO-V3 model. It is observable that our trained YOLO-V3 model detects 7 pomegranates out of 9 present in the input frame.

A. Classifier Implementation

We have implemented our classifier using the fast.ai [8] framework, a PyTorch based deep learning library. We begin training the Resnet50 model with all layers frozen except the attached dense layers at the end. This gives us a baseline model. In Table 5. 1 explained the training loss, valid loss and error rate for the 5 epochs of training with learning rate 0.0005. The accuracy we achieved was 95%.

Table-II: Iteration with Accuracy Classification

Epoch	Train Loss	Valid Loss	Accuracy
1	0.82	0.54	0.78
2	0.69	0.36	0.9
3	0.53	0.24	0.95
4	0.43	0.19	0.95
5	0.36	0.17	0.95

Below is the confusion matrix for the baseline model we trained. A confusion matrix helps us to evaluate the performance of the trained model on a set of data whose target classes are known. We evaluate the model by calculating precision and recall. We obtain True Positives (TP), True Negatives(TN), False Positives(FP), False Negatives(FN) from the confusion matrix below

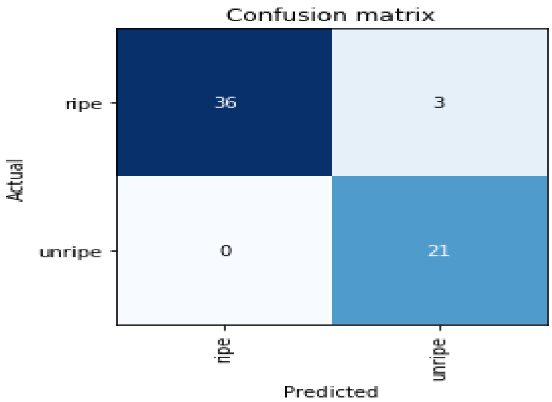

Fig. 7. Confusion matrix with ripe and unripe accuracy

Retrieval Number: C8762019320 /2020©BEIESP
DOI: 10.35940/ijitee.C8762.019320
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

We compute positive predictive value, the proportion of the actual positive identifications from all the positive identifications. It is given by Precision = TP/TP+FP. Likewise, the ratio of positive instances properly predicted from the total number of positive instances that actually existed is given by Sensitivity. It is given by Recall = TP/TP+FN. Both the positive predictive value and sensitivity has equal value of 0.92.

The graph in Fig. 8 shows the changing train and validation losses for the batches processed during the training of the baseline model.

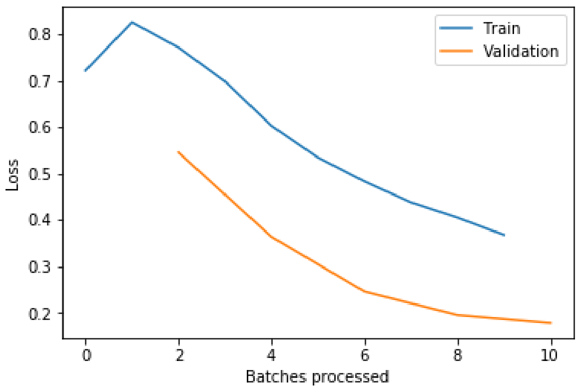

Fig. 8. Results performance with Training vs Validation

Finally, we apply fine-tuning and unfreeze all the frozen layers of the model. To preserve the learned weights of dense layers of the prior trained baseline model, we assigned a small learning rate to this layer compared to initial layers. We set the learning rate to 0.00018 for the initial layers and 0.000055 for the final layers. After training for 5 epochs the accuracy achieved was 98.33 %.

Table-III: Parameters for Experimental analysis

Epoch	Train Loss	Valid Loss	Accuracy
1	0.11	0.16	0.93
2	0.11	0.11	0.96
3	0.09	0.08	0.96
4	0.07	0.06	0.97
5	0.05	0.05	0.98

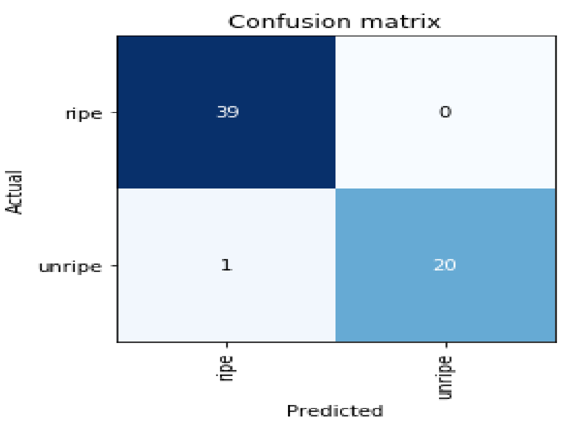

Fig. 9. Confusion matrix obtained from the fine-tuned model

Precision and recall calculated from the confusion matrix for the fine-tuned model are 0.97 and 1 respectively. The graph in Fig. 10 below shows the change in train and validation losses for the batches processed during the fine-tuning of the baseline model. We observe that the train losses and valid losses are approximately equal which signals that our model is not over-fitting or under-fitting on the training data.

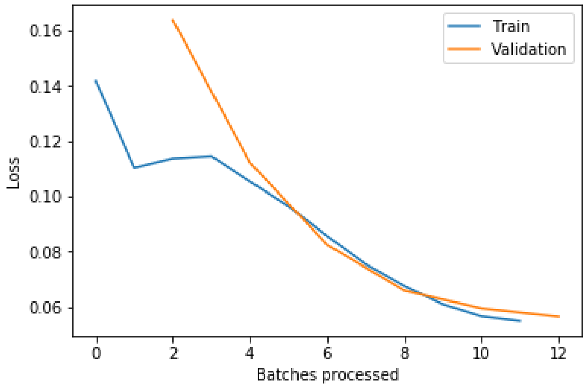

Fig. 10. Change in train and validation losses for the batches processed during the fine-tuning of the baseline model

B. Fruit classification

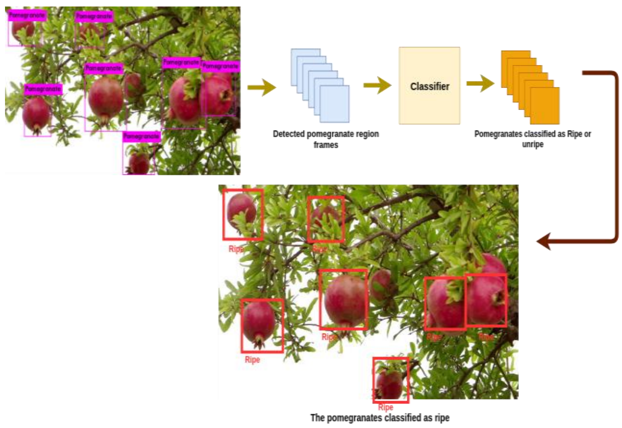

Fig. 11 Results of the proposed model

In the above Fig. 11, we extract the detected pomegranate regions from the YOLO-v3 model output frame and send them to the image classifier we trained earlier to classify them as ripe or unripe. After classification, the respective predicted labels of the detected pomegranate regions are sent back to the original input image and the bounding boxes are renamed from pomegranates to Ripe or unripe according to the label.

V. CONCLUSION AND FUTURE WORK

In this paper, an efficient method of detecting the pomegranate fruits in orchards and classifying them as ripe or unripe is implemented. This proposed work is considered efficient because it first detects the pomegranate from the orchards and sends the detected region into the classifier to classify them as ripe or unripe. Hence, reducing the heavy training costs incurred when only one YOLO-v3 detection model is used to simultaneously detect and classify the pomegranate.

Retrieval Number: C8762019320 /2020©BEIESP
DOI: 10.35940/ijitee.C8762.019320
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

This method is also very flexible as many different classification tasks such as disease detection in pomegranates easily be trained and appended to the model.

REFERENCES

1. S. Bargoti and J. Underwood, "Deep fruit detection in orchards," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 3626-3633.doi: 10.1109/ICRA.2017.7989417
2. K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, “Computer Vision for Fruit Harvesting Robots - state of the Art and Challenges Ahead,” International Journal of Computational Vision and Robotics, vol. 3, no. 1-2, pp. 4–34, apr 2012.
3. A. B. Alexey, “YOLO mark,” 2018. [Online]. Available: https: //github.com/AlexeyAB/Yolo mark
4. Apple detection during different growth stages in orchards using the improved YOLO-V3 model
5. Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. In: IEEE conference on Computer Vision and Pattern Recognition, arXiv:1804.0276.
6. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: unified, realtime object detection. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 779–788.
7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
8. Howard, J., et al. (2018). The fast.ai deep learning library, https://github.com/fastai/fastai.
9. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
10. Koirala, A., Walsh, K.B., Wang, Z. et al. Precision Agric (2019). https://doi.org/10.1007/s11119-019-09642-0
11. Tian, Y., Yang, G., Wang, Z., et al., 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model[J]. Computers and Electronics in Agriculture 157, 417–426.
12. Stajnko, Denis & Lakota, Miran & Hočevar, Marko. (2004). Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. Computers and Electronics in Agriculture. 42. 31-42. 10.1016/S0168-1699(03)00086-3.
13. S. Kim, Y. Ji and K. Lee, "An Effective Sign Language Learning with Object Detection Based ROI Segmentation," 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, 2018, pp. 330-333.
14. Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.
15. 15.Santagapita, P.R., Tylewicz, U. Panarese, V., Rocculi, P.,Dalla, Rosa, M. (2016). Non-destructive assessment of kiwifruit physic-chemical parameters to optimize the osmotic dehydration process, A study on FT-NIR spectroscopy. Journal of Biosyst. Eng. 142, (2), 101-129.
16. 16.Munera, S., Amigo, J. M., Blasco, J.; Cubero, S.; Talens, P., Alexios, N.(2017). Ripeness monitoring of two cultivars of nectarine using VIS-NIR hyperspectral reflectance imaging. Journal of Food Engineering, 214(3), 29-39.
17. 17.Optimal deep learning model for classification of lung cancer on CT images,(2019), Lakshmanaprabu S.K., Sachi Nandan Mohanty, Shankar K., Arunkumar N., Future Generation Computer Systems, 19, 1, 374-382.

AUTHORS PROFILE

Sripad Joshi, completed his bachler’s in Computer Science and Engineering at Faculty of Science and Technology, Icfai Tech, Icfai Foundation for Higher Education(IFHE), Deemed to be University, Hyderabad.

Sandeep Kumar Panda, completed his PhD in Computer Science and Engineering at KIIT University India in 2016. His research interests include Software Engineering, Web Engineering, Cryptography & Security, Blockchain Technology and Cloud Computing. He has published 15 papers in international journals and 10 papers in international conferences. He is currently working as an Associate Professor at ICFAI Foundation for Higher Education (IFHE), Deemed to be University, Hyderabad. His professional affiliations are IEEE, ACM and IAENG.

.

Sathya AR , completed her Master’s degree in Computer Science and Engineering at Anna University , Tamilnadu, India 2007. Her research interests include Software Engineering, Blockchain Technology and Machine Learning. She is currently working as an Assistant Professor at ICFAI Foundation for Higher Education (IFHE), Deemed to be University, Hyderabad.