http://doi.org/10.1590/1678-992X-2022-0064

ISSN 1678-992X

Scientia Agricola

Research article

Detection of coffee fruits on tree branches using computer vision

Helizani Couto Bazame[1]*, José Paulo Molin[1], Daniel Althoff[2], Maurício Martello[1]

[1]Universidade de São Paulo/ESALQ – Depto. de Engenharia de Biossistemas – Lab. de Agricultura de Precisão, Av. Pádua Dias, 11 – 13418-900 – Piracicaba, SP – Brasil.
[2]Stockholm University – Dept. of Physical Geography, Frescativägen, 114 19 – Stockholm – Sweden.
*Corresponding author <helizanicouto@usp.br>

Edited by: Ricardo Enrique Bartosik

Received March 22, 2022
Accepted June 20, 2022

ABSTRACT: Coffee farmers do not have efficient tools to have sufficient and reliable information on the maturation stage of coffee fruits before harvest. In this study, we propose a computer vision system to detect and classify the *Coffea arabica* (L.) on tree branches in three classes: unripe (green), ripe (cherry), and overripe (dry). Based on deep learning algorithms, the computer vision model YOLO (You Only Look Once), was trained on 387 images taken from coffee branches using a smartphone. The YOLOv3 and YOLOv4, and their smaller versions (tiny), were assessed for fruit detection. The YOLOv4 and YOLOv4-tiny showed better performance when compared to YOLOv3, especially when smaller network sizes are considered. The mean average precision (mAP) for a network size of 800 × 800 pixels was equal to 81 %, 79 %, 78 %, and 77 % for YOLOv4, YOLOv4-tiny, YOLOv3, and YOLOv3-tiny, respectively. Despite the similar performance, the YOLOv4 feature extractor was more robust when images had greater object densities and for the detection of unripe fruits, which are generally more difficult to detect due to the color similarity to leaves in the background, partial occlusion by leaves and fruits, and lighting effects. This study shows the potential of computer vision systems based on deep learning to guide the decision-making of coffee farmers in more objective ways.

Keywords: YOLO, precision agriculture, high-quality coffee

Introduction

Coffee demand has increased along with the demand for high-quality products. The supply of high-quality coffee is attributed mainly to improvements in selective harvesting, preferably of ripe fruits (Pineda et al., 2022). Enhancing selective harvest has allowed for the emergence of special products. To meet the increasing demands, new technologies and good crop management practices are needed to improve the quality of harvested coffee without harming the environment.

Most coffee farmers do not have efficient tools to have sufficient and reliable information about the maturation stage of coffee fruits before harvest (Ramos et al., 2017). Tracking the coffee fruits maturation stage can aid the decision of adequate harvesting periods based on the percentage of mature fruits on tree branches (Ramos et al., 2018; Rodríguez et al., 2020). This information is essential for crop management and adequately support decision-making (Martello et al., 2022).

The color of fruit samples is traditionally used to assess the maturation of coffee fruits, and the evaluation can be visual or using colorimeters. Colorimeters measure the color of the fruit surface but without spatial representativeness (Oliveira et al., 2016). Visual classification can also be subjective and relies on the person's experience.

In recent decades, systems based on computer vision have been largely applied to detect and classify fruits (Bazame et al., 2021; Ning et al., 2022; Thendral and David, 2022; Wang et al., 2019; Wu et al., 2020a). Few studies have reported on the classification of coffee fruits before the harvest, which can aid the decision-making of coffee farmers (Avendano et al., 2017; Ramos et al., 2018). However, most of these studies adopted techniques that require first extracting various features and then feeding them to the classification algorithm.

Recent advances in computer vision systems based on deep learning allow several features to be extracted automatically. For example, the YOLO (You Only Look Once) algorithm is a popular computer vision algorithm that has been used in several challenges in agriculture. YOLO has previously been used to detect flowers for robotic pollination (Li et al., 2022), fruit load and maturation (Cuong et al., 2022; Fu et al., 2022; Mirhaji et al., 2021), and weed detection (Parico and Ahamed, 2020). Therefore, this study aims to implement and explore different YOLO algorithms to detect coffee fruits on tree branches and classify the fruits according to the different maturation stages.

Materials and Methods

Data acquisition and labeling

The dataset used in this study consists of 387 RGB images of coffee fruits on tree branches (Figure 1). We used a Smartphone to photograph the fruits before harvest, between 12 and 29 May 2020, from a commercial farm of arabica coffee (Catuaí 144) in the municipality of Patos de Minas, Minas Gerais State, Brazil (18°32′28.55″ S, 46°3′51.17″ W, altitude 1020 m). Although the pictures were taken near the harvest, the crop uneven flowering over time resulted in pictures of coffee fruits with a mix of maturation stages. For developing a robust computer vision model for different field conditions, the pictures were taken from different angles, sides, and plants randomly selected across coffee lines. This resulted in a diverse scenario under different lighting conditions. The pictures were taken without zoom or

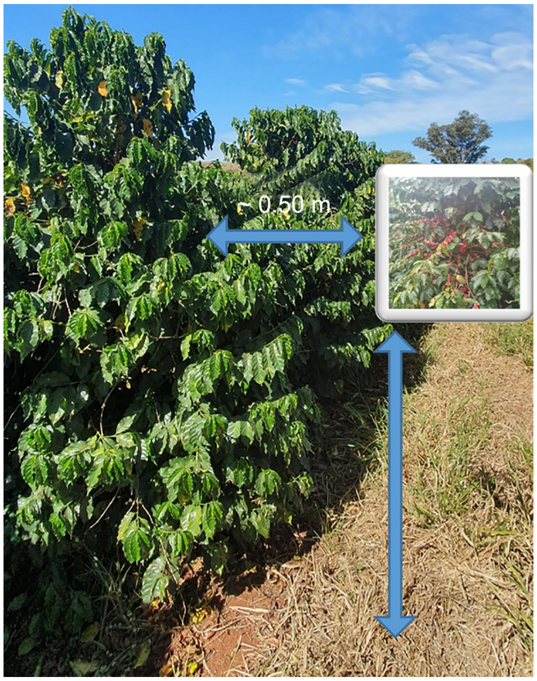

Figure 1 – Image acquisition for coffee fruits on tree branches.

flash and were saved with an image resolution of 72 dpi. The smartphone camera automatically adjusted for the white balance. The images were then randomly split into a training set (~80 % = 310 images) and a testing set (~20 % = 77 images).

The images were annotated considering three stages (classes) of coffee fruit maturation: unripe (green), ripe (cherry), and overripe (or dry). The annotation was carried out using the graphical user interface Yolo Mark (Bochkovskiy et al., 2020).

Computer vision algorithm

This study chose the YOLO algorithm for object detection (Redmon and Farhadi, 2018). The YOLO belongs to a family of one-stage object detectors and is popular for its speed and accuracy (Wu et al., 2020a). In this study, we assessed the improvements of the YOLO latest version, YOLOv4 (Bochkovskiy et al., 2020), compared to its former version, YOLOv3 (Redmon and Farhadi, 2018). The improvements of the YOLOv4 over its former version include using the Mish activation function (Misra, 2019), CutMix and mosaic data augmentation, Cross-Stage Partial connections (CSP), Cross mini-Batch Normalization (CmBN), Spatial Pyramid Pooling (SPP) (He et al., 2015) and the Path Aggregation Network (PANet) blocks, Complete Intersection over Union (CIoU) loss (Zheng et al., 2019), among others.

Besides the YOLOv3 and YOLOv4, a smaller version of these models, termed "tiny", was also assessed. The YOLO-tiny models were developed with fewer convolutional layers and are suitable for constrained devices, such as mobile phones (Tang, 2018), microcomputers, and microcontrollers.

The object detection models were trained considering different network sizes and resampling image sizes to match the corresponding network. The network sizes adopted were 320 × 320, 416 × 416, 512 × 512, 608 × 608, 704 × 704, and 800 × 800 pixels. For training, the batch size was set to 32 in the forward pass and the number of iterations was equal to 6000. The confidence thresholds (c) and non-maximum suppression adopted were 0.25 and 0.45, respectively. The performance criterion was tracked for each training iteration using the test set. The weights with the best performance were adopted as the final weights for the model.

Performance evaluation

The performance of the computer vision algorithms was measured by the mean values of average precisions (mAP) obtained for all classes detected, considering an intersection over union of 50 %. The average precision (Eq. (1)) is the average value of 11 points on the precision/recall curve for pre-determined confidence thresholds for the same class. The precision (Eq. (2)) and recall (Eq. (3)) are computed for 11 equally spaced confidence thresholds (c = 0.0, 0.1, …, 1.0) and precision at each recall level (Eq. (4)) is interpolated by setting the maximum precision measured for a threshold whose corresponding recall r′ exceeds r (Eq. (4)):

$$AP = \int_0^1 p(r)dr \qquad (1)$$

$$p(c) = \frac{TP}{TP + FP} \qquad (2)$$

$$r(c) = \frac{TP}{TP + FN} \qquad (3)$$

$$p(r) = \max_{r':r' \geq r} p(r') \qquad (4)$$

where: AP is the average precision, TP are true positives, FP are false positives, FN are false negatives, $p(r)$ is the precision at recall level r, $p(r')$ is the precision at recall level r', and c is the confidence threshold.

Results and Discussion

The results are presented and discussed in three subsections. The first subsection discusses the general performance obtained by the object detection algorithms and highlights the main findings of this study. The following subsections detail more specific outcomes from the algorithms concerning performance scores for the different classes and object densities, respectively.

General performance obtained by YOLO algorithms

The performance of coffee fruit detection for each YOLO algorithm and network size, as measured by their mean average precision (mAP), is presented in Figure 2. For the YOLOv4, YOLOv4-tiny, and YOLOv3, the mAP stabilized near the network size of 608 × 608 pixels. In contrast, the performance of the YOLOv3-tiny continued to increase until the network size of 704 × 704 pixels. Despite stabilizing, the performance of the algorithms still showed slight improvements up to 800 × 800 pixels. In general, both the YOLOv4 and YOLOv4-tiny outperformed the YOLOv3. The YOLOv4-800 scored the highest mAP (81 %), followed by YOLOv4-tiny-800 (79 %), YOLOv3-800 (78 %), and YOLOv3-tiny-800 (77 %).

The smaller the network size, the greater the YOLOv4 and YOLOv4-tiny outperform the YOLOv3 and YOLOv3-tiny. In contrast, when larger network sizes are considered, for instance, 704 × 704 and 800 × 800 pixels, the difference in performances of the YOLOv3 and YOLOv3-tiny are negligible. Perhaps the most important outcome here is the YOLOv4-tiny outperforming the YOLOv3. This means that the updates made for the latest YOLO version were crucial to improve its performance, even when considering a restricted number of convolutional layers. The YOLOv4-tiny requires ~90 % fewer billion floating-point operations than the YOLOv3, which means its model/weights not only occupy less space in a hard drive but can also be run much faster.

The detections made by the YOLO algorithms for three random images from the dataset and considering the network size of 800 × 800 pixels are shown in Figures 3A, 3B, and 3C. The mAP obtained for each image is also displayed in the figure, where YOLOv4 consistently outperforms the other algorithms. YOLOv4 better detects overlapped fruits (Figure 3C) or in the shade (Figure 3A). It also better detect unripe (green) fruits, even when they are visually smaller in the background and between the leaves (Figure 3B). Another adaptation that could further improve model detection is that suggested by Liu et al. (2020). The authors adapted the YOLO algorithm to use a circular bounding box rather than the traditional rectangular one. Because of the tomato shape, the circular bounding box allowed for better object detection under challenging lighting conditions, branch and leaves occlusion, and overlapping of tomatoes. The proposed algorithm performed better than the other methods and improved detection under occlusion conditions. In Figures 3A, 3B, and 3C, YOLOv4 showed to generally better detect occluded/overlapped objects, even under challenging settings.

The high performance of YOLOv3-tiny (Figure 3A) deserves special attention. There seems to be a surplus of detections (bounding boxes) in the figure, which, despite resulting in high recall (0.92), results in lower precision (0.70) because of the large number of false positives (see Eq. 2 and 3). This is an outcome of the poorly predicted boxes for this specific image not adequately removed by the confidence threshold and non-maximum suppression post-processing. In contrast, the YOLOv3 model predicted coffee fruits in this figure with lower confidence, resulting in fewer boxes and higher precision (0.83), but much lower recall (0.42) and mAP. Despite a similar mAP to that obtained by the YOLOv3-tiny and YOLOv4 models for the example image (Figure 2), YOLOv4 resulted in far better predictions, with both high precision (0.88) and recall (0.88).

To better assess the trade-offs between precision and recall, Figure 4 shows the distribution of performance scores (mAP, precision, and recall) for each test set image. Despite the overall higher median and mean mAP obtained from all images in the test set for YOLOv4, there are clear trends in the precision and recall trade-offs that can be assessed. The mAP is obtained by considering a set of different confidence thresholds, whereas the final precision and recall are calculated assuming a pre-set confidence threshold (c = 0.25). As discussed above, obtaining high precision at the expense of too many false positives can lead to a lower recall. For example, the YOLOv3 algorithm shows, for most network sizes, to score relatively higher precision but lower recall. In contrast, the YOLOv4 algorithm shows the opposite behavior, scoring relatively higher recall and lower precision.

Despite observing general trends for the precision-recall trade-offs for the different algorithms, the results

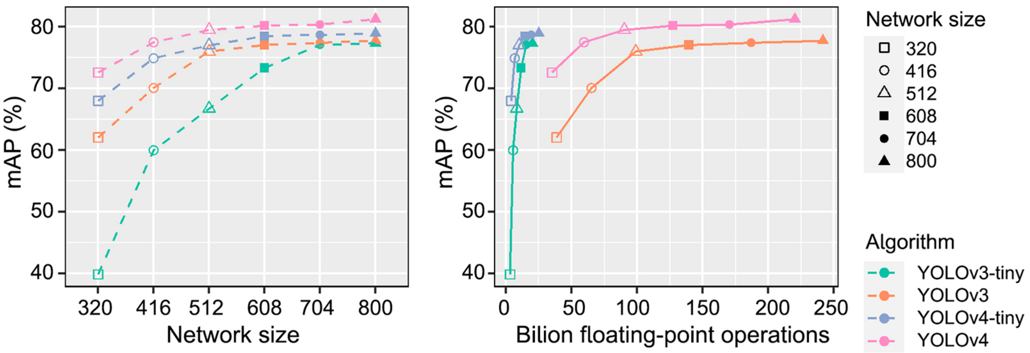

Figure 2 – Performance of the different computer vision algorithms and network sizes assessed to detect coffee fruits on branches. mAP = mean values of average precisions.

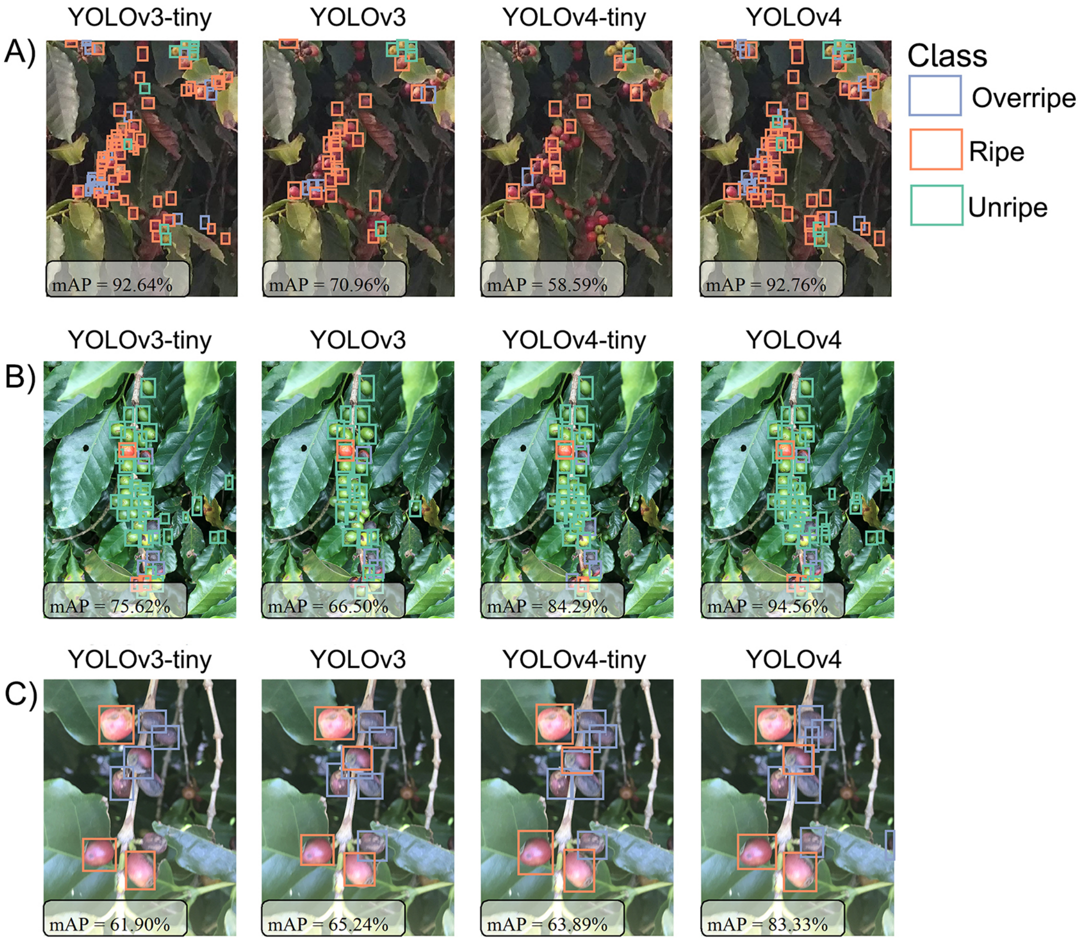

Figure 3 – Coffee fruits detections made by YOLO algorithms considering a network size of 800 × 800 pixels for three arbitrary images representing fruits (A) in the shade, (B) between the leaves, and (C) overlapped.

may partially be attributed to the random weight adjustment process during training. In this study, the final weights of the models were set as the weights obtained after the training iteration that resulted in the highest mAP for the test set from all 6000 iterations. However, predictions from weights scoring similar mAP can present different precision-recall trade-offs. Thus, ultimately, the final user of the model decides whether it is more important to identify all true positives regardless of a few false positives, or if predicting false positives can be detrimental/costly to the final objective. In general, similar values of precision and recall indicate a well-balanced model and a robust precision-recall trade-off.

Performance by detection class

The average precision (AP) obtained for each class highlights a close performance between YOLOv4 and the other models for detecting ripe and overripe coffee fruits, especially for more extensive network sizes (Figure 5). For example, for ripe fruits and a network size of 800 × 800 pixels, the YOLOv4-tiny, YOLO-v3, and YOLOv3-tiny scored APs of 83 %, 84 %, and 80 %, respectively, while YOLOv4 scored an AP (84 %) higher by 1 %, 0.4 %, and 4 %, respectively. For overripe fruits, the YOLOv4-tiny, YOLO-v3, and YOLOv3-tiny scored APs of 78 %, 77 %, and 76 %, respectively, while YOLOv4 scored an AP (80 %) higher by 2 %, 3 %, and 4 %, respectively.

YOLOv4 stands out in detecting unripe (green) coffee fruits, which are generally more difficult to detect because of leaves on the branches and in the background. YOLOv4 scored an AP of 80 % for unripe fruits and a network size of 800 pixels, which is higher by 4 %, 7 %, and 4 % than those scored by YOLOv4-tiny, YOLOv3, and YOLOv3-tiny, respectively. The difference is even higher when smaller network sizes are considered.

Other computer vision systems were also developed to predict the maturation stage of coffee fruits on tree branches (Ramos et al., 2018). The computer vision system classifies coffee fruits after building a 3D model of on-branch coffee fruits and results in classification

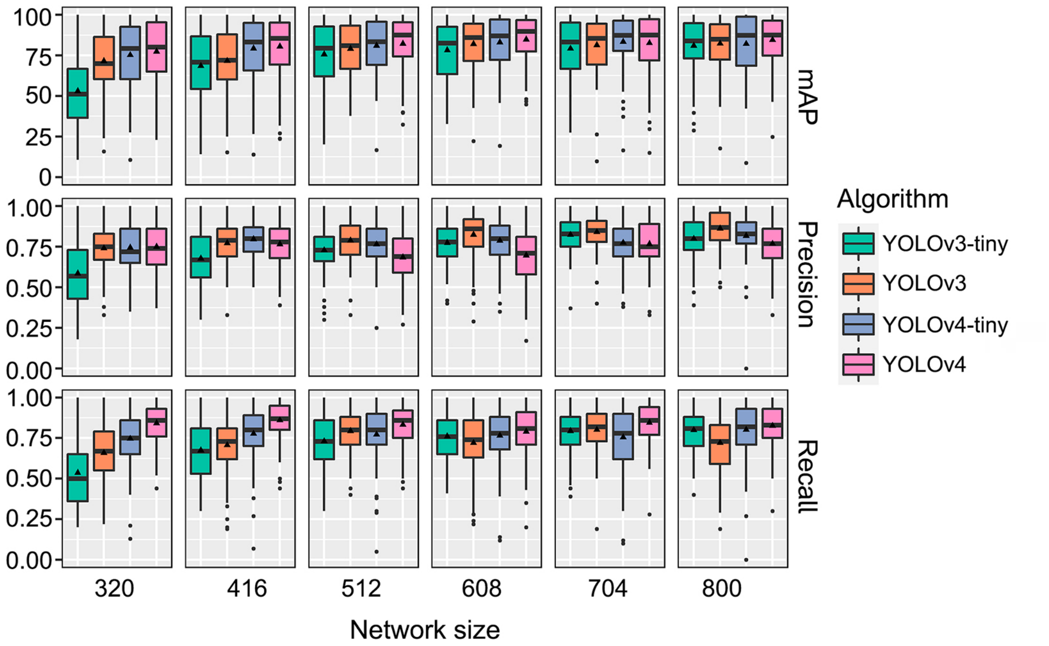

Figure 4 – Distribution of performance scores obtained for each image of the test set by the different computer vision algorithms and network sizes used in this study. mAP = mean values of average precisions.

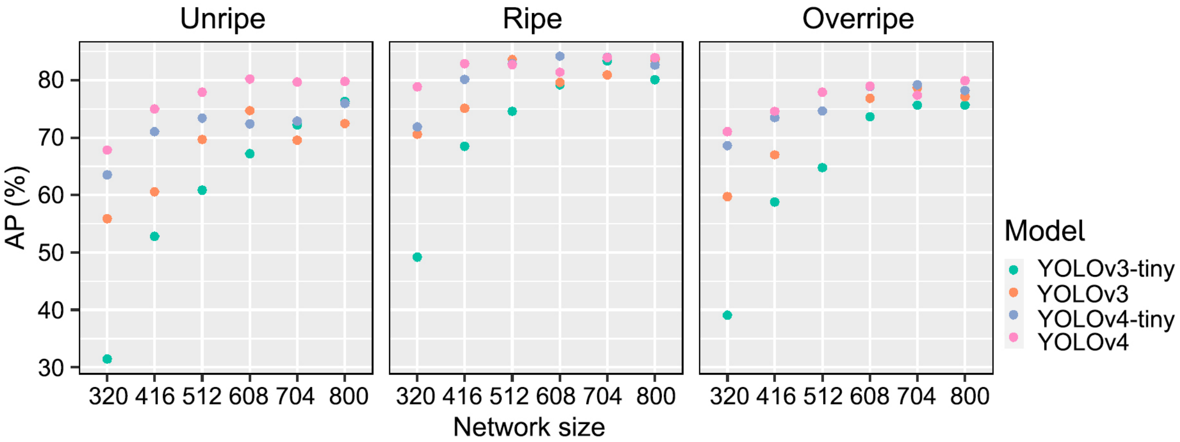

Figure 5 – Performance of the different computer vision algorithms and network sizes assessed for each class of detection. AP = Average precision.

efficacy between 42 % and 92 % for the different classes of the maturation stage. A computer vision model to detect coffee fruits and classify their maturation stage during harvest was proposed by Bazame et al. (2021). The authors then mapped the maturation stage across the coffee plantation with an mAP of 86 %, 85 %, and 80 % for unripe, ripe, and overripe fruits, respectively. The lower mAP for unripe fruits in this study, compared to that of Bazame et al. (2021), can be attributed to the environment where images were taken. Here, pictures were taken from on-branches coffee fruits with a diverse background, including leaves and shades, whereas Bazame et al. (2021) collected data inside the harvester where the environment had controlled illumination and contrasting background. Besides, the authors also registered a lower mAP score for overripe fruits.

A further opportunity for the present study could be related to predicting coffee yield from full lateral pictures of coffee plants, as proposed by Idol and Youkhana (2020). However, obtaining such information for field scales requires collecting images along with geographic coordinates at higher rates. Besides, data collection at higher rates by autonomous systems has been proposed in different studies. For example, an autonomous robot to monitor vineyard water potential was proposed by Saiz-Rubio et al. (2021). Autonomous robots have even been proposed to perform actions, such as tomato harvesting (Liu et al., 2020), strawberry harvesting (Xiong et al., 2020), and weed control (Wu et al., 2020b).

Performance for different object densities

It is harder for a smaller network to detect coffee fruits in higher object-density scenarios. This is because resizing images to lower resolution may blur the boundaries of fruits. This behavior is evident in Figure 6, which shows lower median mAP (red dashed lines) obtained for smaller networks and steeper slopes for the ordinary least squares regression fitted to data (blue line). For example, the YOLOv3 and YOLOv3-tiny models resulted in mAP lower than 70 % and 57 %, respectively, in 50 % of the images in the test set for a network size of 320 × 320 pixels. YOLOv4-tiny and YOLOv4 were more robust to extract features and avoid these effects for the smaller network sizes. For YOLOv4-tiny and YOLOv-4, 50 % of the test set images scored mAP equal to or higher than

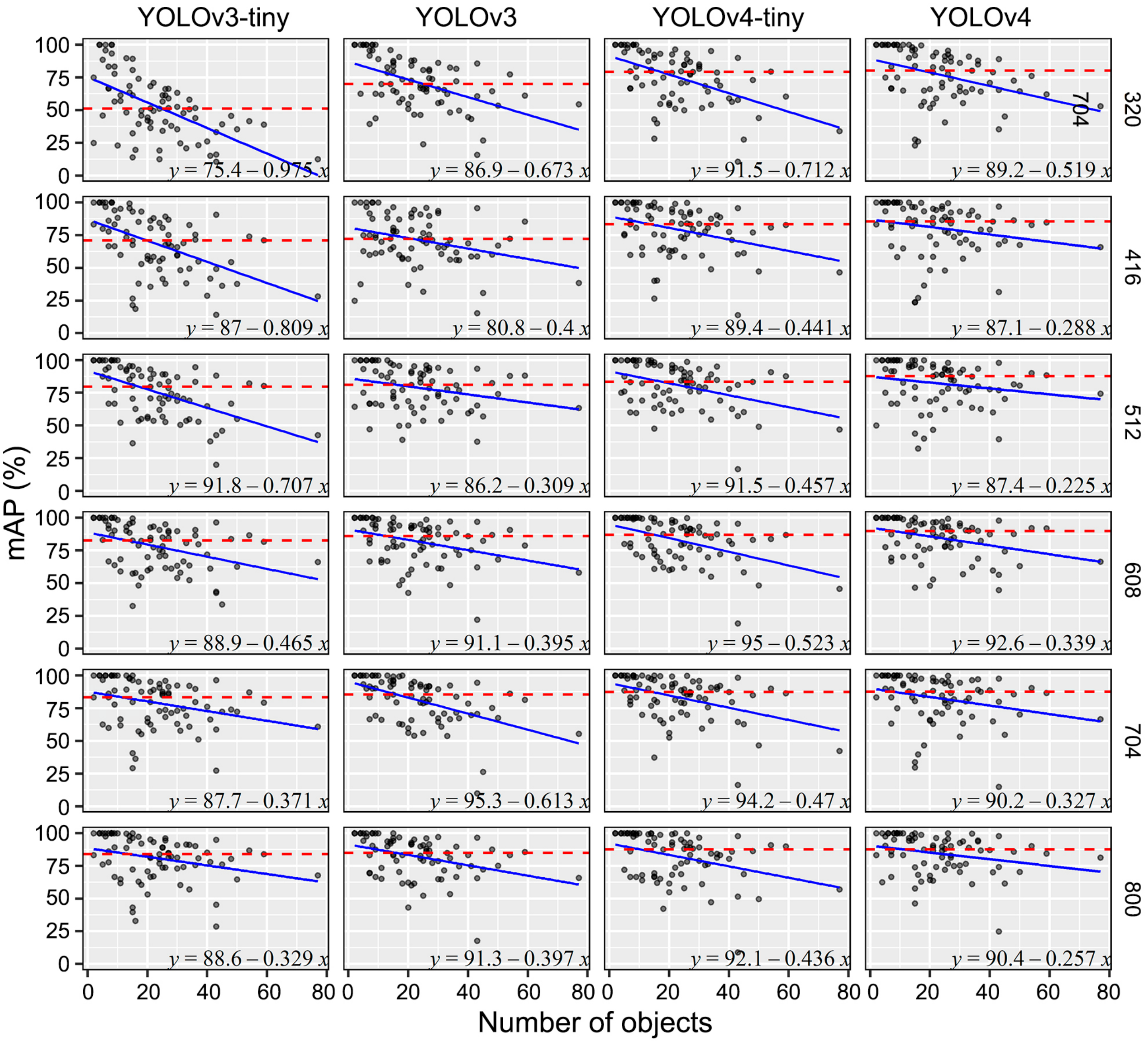

Figure 6 – Performance obtained by the different computer vision algorithms and network sizes assessed for each image of the test set separately. The red dashed line represents the median mAP. The blue line represents the ordinary least squares regression fit to the data. Steeper slopes mean that it is more difficult for the model to detect objects when object density is higher in the dataset. mAP = mean values of average precisions.

79 % and 80 % for a network size of 320 × 320 pixels. An adaption to the YOLOv3 model for the detection of litchi (YOLOv3-Litchi) in images with a high density of fruits has been proposed by Wang et al. (2021). The authors adapted the model to have fewer convolutions than the original YOLOv3 and predict from feature maps at higher resolutions, which increased accuracy to detect objects in images with high densities of small fruits.

As the network size and, therefore, the resolution of resized images increases, the problem is mitigated. For example, the regression slopes for the YOLOv3-tiny models decreased from −0.975 to −0.329 for network sizes from 320 to 800 × 800 pixels. Overall, the regressions adjusted more gentle slopes (closer to 0) for scores obtained using larger network sizes. This is especially true for the YOLOv4 algorithm, whose slope was only −0.257 for the network size of 800 × 800 pixels. Input images at higher resolutions mean more extensive networks and usually better performance in object detection, but it may also increase the time required to predict (Wang et al., 2021) or constrain the model to hardware with higher computing power. The YOLOv4-tiny also performed better than YOLOv3-tiny in this regard, even at smaller network sizes, which can be attributed to its more robust feature extractor.

Conclusions

In this study, the YOLOv3 and YOLOv4 object detection algorithms were implemented to detect and classify the maturation stage of coffee fruits on tree branches. For an image input resolution of 320 × 320 pixels, YOLOv4, YOLOv4-tiny, YOLOv3, and YOLOv3-tiny scored a mean average precision (mAP) of 73 %, 68 %, 62 %, and 40 %, respectively. For larger networks, considering images of 800 × 800 pixels, these models scored mAPs of 81 %, 79 %, 78 %, and 77 %, respectively.

The developed models better detect ripe coffee fruits, which better contrast the background of the images. In contrast, the performance to detect unripe (green) fruits was considerably lower, which can be attributed to the coffee fruits being partially occluded by leaves (similar color) and in the shade. Overall, the YOLOv4 algorithm was more robust into detecting unripe fruits and less influenced by object density in images.

Future studies could advance this research in many directions. The image acquisition could be associated with geographic coordinates or even captured by an automated system, allowing for the spatialization of such information. The continuous collection of images from all sides of coffee plants could also be used to estimate fruit count and therefore,plant yield.

Acknowledgments

To the Guima Café Group for the experimental area and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for granting the scholarship of authors 1 and 4 (Finance Code 001).

Authors' Contributions

Conceptualization: Bazame, H.C.; Molin, J.P. **Formal analysis:** Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M. **Investigation:** Bazame, H.C.; Althoff, D. **Methodology:** Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M. **Project administration:** Molin, J.P. **Software:** Bazame, H.C.; Althoff, D. **Validation:** Bazame, H.C.; Martello, M. **Visualization:** Bazame, H.C.; Althoff, D.; Martello, M. **Writing-original draft:** Bazame, H.C. **Writing-review & editing:** Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M.

References

Avendano, J.; Ramos, P.J.; Prieto, F.A. 2017. A system for classifying vegetative structures on coffee branches based on videos recorded in the field by a mobile device. Expert Systems with Applications 88: 178-92. https://doi.org/10.1016/j.eswa.2017.06.044

Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M. 2021. Detection, classification, and mapping of coffee fruits during harvest with computer vision. Computers and Electronics in Agriculture 183: 106066. https://doi.org/10.1016/j.compag.2021.106066

Bochkovskiy, A.; Wang, C.; Liao, H.M. 2020. YOLOv4: Optimal speed and accuracy of object detection. Available at: http://arxiv.org/abs/2004.10934 [Accessed Oct 25, 2021]. https://doi.org/10.48550/arXiv.2004.10934

Cuong, N.H.H.; Trinh, T.H.; Meesad, P.; Nguyen, T.T. 2022. Improved YOLO object detection algorithm to detect ripe pineapple phase. Journal of Intelligent & Fuzzy Systems 43: 1365-1381. https://doi.org/10.3233/JIFS-213251

Fu, L.; Yang, Z.; Wu, F.; Zou, X.; Lin, J.; Cao, Y.; Duan, J. 2022. YOLO-Banana: a lightweight neural network for rapid detection of banana bunches and stalks in the natural environment. Agronomy 12: 391. https://doi.org/10.3390/agronomy12020391

He, K.; Zhang, X.; Ren, S.; Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37: 1904-16. https:// doi.org/10.1109/TPAMI.2015.2389824

Idol, T.W.; Youkhana, A.H. 2020. A rapid visual estimation of fruits per lateral to predict coffee yield in Hawaii. Agroforestry Systems 94: 81-93. https://doi.org/10.1007/s10457-019-00370-y

Li, G.; Suo, R.; Zhao, G.; Gao, C.; Fu, L.; Shi, F.; Dhupia, J.; Li, R.; Cui, Y. 2022. Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. Computers and Electronics in Agriculture. 193: 106641. https://doi.org/10.1016/j.compag.2021.106641

Liu, G.; Nouaze, J.C.; Mbouembe, P.L.T.; Kim, J.H. 2020. YOLO-Tomato: a robust algorithm for tomato detection based on YOLOv3. Sensors 20: 21-45. https://doi.org/10.3390/S20072145

Martello, M.; Molin, J.P.; Bazame, H.C. 2022. Obtaining and validating high-density coffee yield data. Horticulturae 8: 421. https://doi.org/10.3390/horticulturae8050421

Mirhaji, H.; Soleymani, M.; Asakereh, A.; Mehdizadeh, S.A. 2021. Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. Computers and Electronics in Agriculture 191: 106533. https://doi.org/10.1016/j.compag.2021.106533

Misra, D. 2019. Mish: a self regularized non-monotonic activation function. *ArXiv*. Available at: http://arxiv.org/abs/1908.08681 [Accessed Nov 10, 2021]

Ning, Z.; Luo, L.; Ding, X.; Dong, Z.; Yang, B.; Cai, J.; Chen, W.; Lu, Q. 2022. Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards. Computers and Electronics in Agriculture 196: 106878. https://doi.org/10.1016/j.compag.2022.106878

Oliveira, E.M.; Leme, D.S.; Barbosa, B.H.G.; Rodarte, M.P.; Pereira, R.G.F.A. 2016. A Computer vision system for coffee beans classification based on computational intelligence techniques. Journal of Food Engineering 171: 22-27. https://doi.org/10.1016/j.jfoodeng.2015.10.009

Parico, A.I.B.; Ahamed, T. 2020. An aerial weed detection system for green onion crops using the you only look once (YOLOv3) deep learning algorithm. Engineering in Agriculture, Environment and Food 13: 42-48. https://doi.org/10.37221/eaef.13.2_42

Pineda, M.F.; Tinoco, H.A.; Lopez-Guzman, J.; Perdomo-Hurtado, L.; Cardona, C.I.; Rincon-Jimenez, A.; Betancur-Herrera, N. 2022. Ripening stage classification of *Coffea arabica* L. var. Castillo using a machine learning approach with the electromechanical impedance measurements of a contact device. Materialstoday: Proceedings 62: 6671-6678. https://doi.org/10.1016/j.matpr.2022.04.669

Ramos, P.J.; Prieto, F.A.; Montoya, E.C.; Oliveros, C.E. 2017. Automatic fruit count on coffee branches using computer vision. Computers and Electronics in Agriculture 137: 9-22. https://doi.org/10.1016/j.compag.2017.03.010

Ramos, P.J.; Avendaño, J.; Prieto, F.A. 2018. Measurement of the ripening rate on coffee branches by using 3D images in outdoor environments. Computers in Industry 99: 83-95. https://doi.org/10.1016/j.compind.2018.03.024

Redmon, J.; Farhadi, A. 2018. YOLO v.3: An Incremental Improvement. Cornell University, Ithaca, NY, USA. (Technical Report).

Rodríguez, J.P.; Corrales, D.C.; Aubertot, J.N.; Corrales, J.C. 2020. A computer vision system for automatic cherry beans detection on coffee trees. Pattern Recognition Letters 136: 142-153. https://doi.org/10.1016/j.patrec.2020.05.034

Saiz-Rubio, V.; Rovira-Más, F.; Cuenca-Cuenca, A.; Alves, F. 2021. Robotics-based vineyard water potential monitoring at high resolution. Computers and Electronics in Agriculture 187: 106311. https://doi.org/10.1016/j.compag.2021.106311

Tang, J. 2018. GitHub - Jeffxtang/Yolov2_tf_ios: Object detection with YOLOv2 and tensorFlow on IOS. Available at: https://github.com/jeffxtang/yolov2_tf_ios#readme [Accessed Nov 20, 2021]

Thendral, R.; David, D.S. 2022. An enhanced computer vision algorithm for apple fruit yield estimation in an orchard. Artificial Intelligence and Technologies 806: 263-273. https://doi.org/10.1007/978-981-16-6448-9_27

Wang, H.; Dong, L.; Zhou, H.; Luo, L.; Lin, G.; Wu, J.; Tang, Y. 2021. YOLOv3-Litchi detection method of densely distributed litchi in large vision scenes. Mathematical Problems in Engineering 2021:1-11. https://doi.org/10.1155/2021/8883015

Wang, Z.; Walsh, K.; Koirala, A. 2019. Mango fruit load estimation using a video based MangoYOLO-Kalman filter-hungarian algorithm method. Sensors 19: 2742. https://doi.org/10.3390/s19122742

Wu, D.; Lv, S.; Jiang, M.; Song, H. 2020a. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture 178: 105742. https://doi.org/10.1016/j.compag.2020.105742

Wu, X.; Aravecchia, S.; Lottes, P.; Stachniss, C.; Pradalier, C. 2020b. Robotic weed control using automated weed and crop classification. Journal of Field Robotics 37: 322-340. https://doi.org/10.1002/rob.21938

Xiong, Y.; Ge, Y.; Grimstad, L.; From, P.J. 2020. An autonomous strawberry-harvesting robot: design, development, integration, and field evaluation. Journal of Field Robotics 37: 202-224. https://doi.org/10.1002/rob.21889

Zheng, Z.; Wang, P.; Liu, W.; Li, J.;Ye, R.; Ren, D. 2019. Distance-IoU loss: faster and better learning for bounding box regression. ArXiv. Available at: http://arxiv.org/abs/1911.08287 [Accessed Dec 05, 2021]