A Novel Approach to Cashew Nut Detection in Packaging and Quality Inspection Lines

Van-Hung Pham, Ngoc-Khoat Nguyen*,Van-Minh Pham
Faculty of Electrical Engineering, Hanoi University of Industry, Hanoi, Vietnam[1, 3]
Faculty of Control and Automation, Electric Power University, Hanoi, Vietnam[2]

***Abstract*—YOLO standing for *You Only Look Once* is one of the most famous algorithms in computer vision used for detecting objects in a real-time environment. The newest version of this algorithm, namely YOLO with the seventh version or YOLOv7, is proposed in the present study for cashew nut detection (good, broken and not peeled) in packaging and quality inspection lines. Furthermore, this algorithm using an efficient convolutional neural network (CNN) to be able to successfully detect and identify unsatisfactory cashew nuts, such as chipped or burnt cashews. In order to deal with the quality inspection process, a new dataset called CASHEW dataset has been built at first by collecting cashew images in environments with different brightness and camera angles to ensure the model's effectiveness. The quality inspection of cashew nuts is tested with a huge number of YOLOv7 models and their effectiveness will also be evaluated. The experimental results show that all models are able to obtain high accuracy. Among them, the YOLOv7-tiny model employs the least number of parameters, i.e. 6.2M but has many output parameters with higher accuracy than that of some other YOLO models. As a result, the proposed approach should clearly be one of the most feasible solutions for the cashew's quality inspection.**

Keywords—Cashew; CNN; cashew detection; YOLOv7; computer vision

I. INTRODUCTION

It is undeniable cashews (anacardium occidentale) are among the most widely consumed nuts globally, owing to their nutritional value. As of 2016, the world's cashew nut production reached 4.89 million tons, with Vietnam being the leading producer [1]. Cashews are high in protein and polyunsaturated fats, high in carbohydrates and fats, and contain abundant calcium, iron, and phosphorus [2]. Often, they are served as snacks or incorporated into confectioneries or food preparations to increase their value and are generally consumed in raw, roasted, salted or flavoured forms. Consequently, cashews provide various health benefits, such as cancer prevention, cardiovascular protection, nerve protection, antioxidant action, and vitamin content [1].

The cashew nut processing process includes many stages such as roasting, shell cracking and removal, peeling of kernel skin, grading, and packaging [3]. The main issue affecting product cost in the cashew nut processing industry is the yield and quality of cashew nuts after processing. Therefore, many automation technologies have been applied in the cashew nut production process. However, cashew nuts are often graded by vibrators, removing unsatisfactory particles, followed by a manual inspection and hand-picking. In general, skilled workers grade cashews based on their size, shape, and color. Such a manual inspection method is time-consuming, tedious, laborious and less productive.

Therefore, several studies have applied a computer vision approach to assist people with classification, defect detection, quality inspection, and grading of fruits [4-9], vegetables [10-12], grains [13-15], and other food products [16, 17]. For example, Cervantes-Jilaja et al. [13] proposed a computer vision-based method to detect and identify visual defects in chestnuts using external features such as shape, color, size, and texture. In [18], the authors used invertible neural networks to locate and segment damaged soybean seeds in a fast and effective manner. Several review papers on the classification and quality evaluation of fruits and vegetables based on a computer vision approach can be found in [12, 19].

This paper aims to propose a novel algorithm for detecting and classifying cashew nuts. A practical system is also built to verify the studied method in steads of using a manual system. The main contributions of this study focus on:

1) Proposing a new algorithm based on an efficient convolutional neural network (CNN) to detect and recognize unsatisfactory cashew nuts, i.e. chipped or burnt cashew ones.

2) Designing a simulation model representing practical packaging and inspection lines systems consists of conveyor and camera (see Fig. 1). This model is employed to construct a dataset of cashew images.

3) Using the newest and best version of YOLO (v7-tiny) to effectively train the models and evaluate quality of these models.

The rest of this paper is organized as follows. First, Section II presents a number of related works which motivate this study. Then, Section III presents the materials and methods for cashew nut detection and recognition. Next, Section IV shows the evaluation results, while the last section provides the study's conclusions and future work.

II. RELATED WORKS

There have been some computer vision-based studies regarding the identification and grading of cashew nuts in the literature. The simplest methods of classifying cashew kernels are based on their color or texture and a single layer neural network [20, 21]. Aran et al. [22] extended this work by testing external features like color, texture, shape and size. In addition, they analyzed the impact of some image preprocessing methods on cashew nut classification. Finally, they evaluated

*Corresponding author: Ngoc-Khoat Nguyen

several classifiers in terms of accuracy and performance. The results show that Back Propagation Neural Network (BPNN) was the most optimal among the classifiers. However, these studies will not be able to classify splits, which occur when kernels split lengthwise naturally. In [23], a novel machine vision method for classifying whole and split cashews (split-up and split-down) is proposed using object shadow combined with surface grayscale-intensity profile.

In addition, methods based on convolutional neural networks (CNNs) are also applied to classify cashew nuts. For instance, Shivaranjani et al. [15] introduced CashNet-15: an optimized cashew nut grading system based on deep CNN and data augmentation. However, this method can only classify wholes and others (scorched whole, splits, butts, pieces). In [24], four deep CNNs combined with image processing were used to classify cashew kernels into five categories based on their adulteration with butts and pieces. Therefore, the above studies demonstrate the effectiveness of machine vision in the classification and grading of cashew nuts. To the best of our knowledge, no study has been conducted on the cashew detection problem, i.e. being able to identify the location and type of cashews (good, broken, and shelled) as they move along the packaging and product inspection lines. Solving this problem can help remove bad nuts (broken or shelled) before packaging and automate product quality inspection.

In recent years, CNN-based object detection has made significant progress and applied to many practical problems. There are two main approaches. The first is a two-stage target detection method such as Faster Region-based CNN (Faster R-CNN) [25], which identifies proposed regions first and then classify them by region. The other method employs a single neural network to detect the type and location of objects simultaneously in an image, such as Single shot multi-box detector (SSD) [26] and YOLO [27-30]. Due to its much faster execution speed than Faster R-CNN, the YOLO model is continuously improved in both speed and accuracy of detection, with many versions such as YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5 [27-30] and YOLOv7 [31]. Several studies have been conducted using Faster R-CNN [14, 32, 33] or YOLO [34-38] to detect fruits.

However, these approaches have not been applied to the cashew detection problem. This paper proposes an algorithm based on a convolutional neural network (CNN) to detect and identify unsatisfactory cashew nuts, such as chipped or burnt cashews. First, we construct a dataset containing cashew images obtained from cameras mounted on packaging and inspection lines, and then we label the images. Then, we use the YOLOv7 (version 7) to train the model on this dataset. Finally, we conduct a quantitative evaluation of the proposed model's effectiveness on the test dataset and a qualitative assessment from the videos obtained by cameras mounted on the cashew packing line.

III. MATERIALS AND METHODS

A. YOLOv7

YOLO (You Only Look Once) is a well-known CNN architecture used for general object detection problems because it balances quality requirements and speed. This architecture not only detects the presence of an object but also determines the object's position in an image. YOLO has three official versions YOLOv1, YOLOv2, and YOLOv3. Then there are many improved versions e.g. YOLOv4, YOLOv5, YOLOX [39], YOLOR [40], and the latest is YOLOv7. The newest version, YOLOv7, significantly enhanced speed and accuracy over previous versions. Therefore, in this paper, we choose this version for cashew detection.

YOLOv7 has applied several improvements to increase speed and accuracy. The computational block in the YOLOv7 backbone is called E-ELAN, which stands for Extended Efficient Layer Aggregation Network. Through the E-ELAN architecture of YOLOv7, the network can continuously improve its learning ability by using "expand, shuffle, merge cardinality" without losing its gradient path.

Model scaling allows the generation of models that meet the requirements of different applications by adjusting key attributes. It is possible to optimize a model by scaling it in terms of width (number of channels), depth (number of stages), and resolution (input image size). With traditional concatenation-based architectures (for example, ResNet or PlainNet), scaling factors cannot be analyzed independently and must be analyzed together. Model depth scaling, for example, causes a change in the ratio between the input and output channel of a transition layer, resulting in a decrease in hardware usage. As a result, YOLOv7 introduces a compound model scaling method for concatenation-based models. Compound scaling maintains the model's properties at its design level and thus maintains its optimal structure. Scaling compound models involves two steps: scaling the depth factor of a computational block requires a change in its output channel, and width factor scaling requires a similar change in its transition layer.

RepConv achieves excellent performance in VGG architectures but is significantly less accurate when applied directly to ResNets and DenseNets. YOLOv7 implements re-parameterized convolution using RepConv without identity connection (RepConvN). Re-parameterized convolution avoids an identity connection when replacing residual or concatenation convolution with a re-parameterized one.

A YOLO architecture contains a backbone, a neck, and a head. The head responsible for the final output is called the lead head, and the head used to assist training in the middle layers is named the auxiliary head. In addition, and to enhance the deep network training, a Label Assigner mechanism was introduced that considers network prediction results together with the ground truth and then assigns soft labels. Compared to traditional label assignment that uses ground truth as a basis for generating hard labels based on given rules, reliable soft labels utilize calculation and optimization methods that also consider the quality and distribution of prediction outputs along with the ground truth.

B. Dataset

To collect the dataset, we built a simulation model for the actual packaging and quality inspection lines, as shown in Fig. 1.

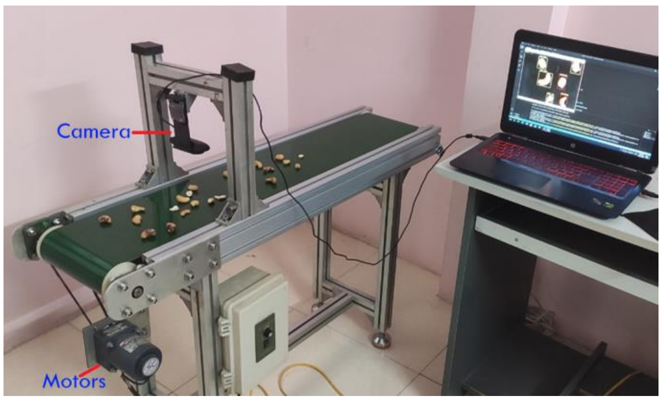


Fig. 1. Cashew nut identification system model

A surveillance camera is securely mounted on a rack and viewed at right angles to the conveyor belt. The camera is connected to a computer via a USB port to capture images of cashews moving on the conveyor. To ensure the robustness of the model in practical applications, we changed the light intensity of the laboratory to collect images of cashew nuts at different brightness levels (strong, normal, and weak). Furthermore, the cashews are placed on the conveyor with varying rotation angles and distances from the camera's position. In this study, we only consider three types of cashew nuts that need to be detected and classified: broken cashews, unshelled cashews, and good cashews (not broken and peeled), as shown in Fig. 2.

Fig. 2. Types of cashews to detect and classify: good cashews (left), broken cashews (middle) and not peeled cashews (right)

After collecting the images, the next step is to annotate them manually using the graphical image annotation tool LabelImg [41]. The annotation classes include 0-good, 1-broken, and 2-Not peeled. As a result, a new dataset named CASHEW dataset was constructed. It contains 312 images and 5115 instances, of which 80% (248 images) were selected randomly as the training dataset, 10% (32 images) as the validation dataset, and the remaining 10% (32 images) as the test dataset. The test dataset was only used to evaluate the model performance after the training, as shown in Table I.

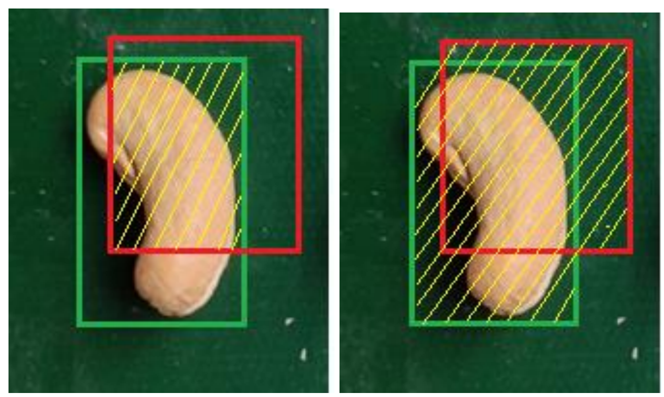
TABLE I. NUMBER OF ANNOTATED IMAGES FOR EACH CASHEW TYPE

CASHEW dataset	*Good*	*Broken*	*Not peeled*	*Number of images*
Train	*1387*	*1569*	*1163*	*248*
Test	*316*	*139*	*89*	*32*
Val	*154*	*164*	*134*	*32*

C. Metrics for Performance Evaluation

To evaluate the effectiveness of a model on the test dataset, we use the following metrics: precision (P), recall (R), average precision (AP), and mean average precision (mAP). Before defining these metrics, we need to define an intersection over union (IoU). The IoU is defined as the percentage overlap between the ground truth (G) and detection boxes (D) and is calculated as:

$$IoU = \frac{Intersection}{Union} = \frac{G \cap D}{G \cup D} \tag{1}$$



Intersection Union

Fig. 3. Examples of intersection and union. ground truth (green) and detection (red)

An example describing the intersection and union with ground truth (green) and detection (red) is shown in Fig. 3. For quantifying accuracy, confusion matrix criteria are calculated, such as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP should have the same ground truth class as detection, and its IoU should be greater than 50%. In contrast, if the detection belongs to the same class as the ground truth and its IoU is less than 50%, it is considered FP. If a ground truth box exists, but the model does not make any detection, this situation is categorized as FN. Lastly, there is no ground truth or detection for the background, so it is classified as TN.

Precision refers to the percentage of correct positive predictions among all detections, while Recall represents the percentage of true positives among all ground truths, as shown in (2) and (3).

$$P = \frac{TP}{TP+FP} = \frac{TP}{\text{All detection}} \quad (2)$$

$$R = \frac{TP}{TP+FN} = \frac{TP}{\text{All ground truths}} \quad (3)$$

The average precision (AP) is the area under the Precision multiplies Recall curve for each class. This can be calculated using (4). After this, the mean average precision (mAP), presenting the overall performance of the model in the detection of possible classes, could be estimated by calculating the mean of all classes' AP.

$$AP = \sum_{i=1}^{n} p(i).\Delta r(i) \quad (4)$$

Where n indicates the total number of detections, i refers to the rank of detection in the list of sorted detections, p(i) represents the precision of the sub-list from the first to the ith detection, and $r(i)$ represents the change in Recall from $(i-1)th$ to ith detection.

IV. Results and Discussions

A. Dataset Training of YOLOv7

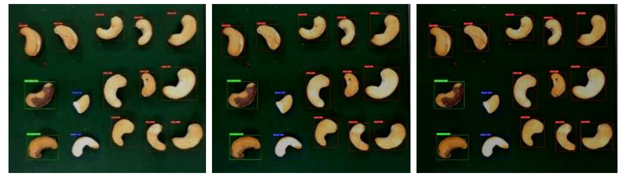
We next conduct the training on the Google Collab Pro platform with the dataset described above. YOLOv7 is trained on the Pytorch platform. First, we select the desired model and edit the configuration file to match the requirements. Next, based on the Train and Val datasets, use pre-trained weights to conduct training process. The number of layers to be classified is 3, including 0-good, 1-broken, and 2-Not peeled. The stochastic gradient descent (SGD) was adopted with an initial learning rate of 0.001. The hyper-parameters used in training the YOLO models are shown in Table II.

TABLE II. Hyper-Parameters Used in Training YOLOv7 Models

Models	Initial Learning Rate	Momentum	Decay	Batch Size
YOLOv7-tiny	0.01	0.937	0.0005	16
YOLOv7	0.01	0.937	0.0005	8
YOLOv7X, W6, E6	0.01	0.937	0.0005	4
YOLOv7D6, E6E	0.01	0.937	0.0005	2

B. Results of Evaluation

Some results of detecting cashews while moving on a conveyor belt with the YOLOv7 model are shown in Fig. 4. It is found that, due to the reflection of light, the accurate detection of cashew nuts is good, chipped or left peeled easily confused. We have tested the performance of the proposed model under different ambient light conditions. The results in Fig. 4 show the same case with three other lighting conditions (strong, normal, and weak). YOLOv7 model could still detect cashews, whether they were unshelled or broken, even though the particles were traveling at less than 0.6 cm/s on the conveyor.



Strong light Normal light Weak light

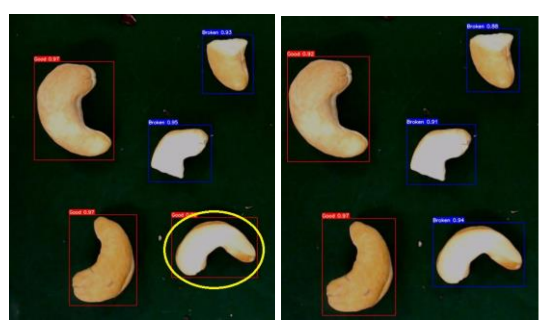
Fig. 4. Results of the YOLOv7 model for detecting cashew nuts as they move on a conveyor at a speed of 0.6 cm/s. The algorithm can accurately detect not_peeled (green), broken (blue), and good seeds (red)

Next, we perform a quantitative evaluation of the model's effectiveness on the test dataset. The evaluation results are given in Table III below. The YOLOv7tiny, YOLOv7, and YOLOv7X models with 6.2M, 36.9M, and 35.7M parameters, respectively, were trained with an input image of size 640 × 640. YOLOv7tiny model has the least number of parameters but has higher F1-score, Precision, Recall, and mAP values than YOLOv7. Even YOLOv7tiny has higher F1-score and Recall parameters than YOLOv7X.

The YOLOv7W6, YOLOv7E6, YOLOv7D6, and YOLOv7E6E models had 70.4M, 97.2M, 154.7M, 151.7M, respectively, and were all trained with 1280×1280 input images. Model YOLOv7D6 has the highest number of parameters, 154.7M; however, many parameters, like Precision and mAP, have the smallest value (blue) among all models.

Comparing all seven models shows that YOLOv7W6 has the largest Recall and mAP values, while YOLOv7E6E model has the largest Precision and F1-score values. On the other hand, we also find that the YOLOv7tiny model with the least number of parameters can be installed on low-profile embedded computers but can still achieve high evaluation indicators. Therefore, the YOLOv7tiny model can be applied to detect and identify actual cashew nuts.

When considering the influence of the ambient light intensity, we found that the light intensity can affect the model's detection results. In Fig. 5, the YOLOv7E6E model incorrectly detected one case (highlighted in yellow), while the YOLOv7tiny model with much fewer parameters correctly detected all the cases.



(a) Detection result of YOLOv7E6E (b) Detection result of YOLOv7tiny

Fig. 5. Comparing YOLOv7tiny and YOLOv7E6E cashew detection results in weak light conditions

TABLE IV. ENT RESULTS ON THE TEST DATASET

Model	#Param	Image size	F1-score	Precision	Recall	AP			mAP
						Good	*Not peeled*	*Broken*	
YOLOv7tiny	6.2M	640	0.88	0.88	0.88	0.952	0.942	0.824	0.906
YOLOv7	36.9M	640	0.86	0.862	0.856	0.934	0.938	0.827	0.900
YOLOv7X	71.3M	640	0.84	0.881	0.801	0.960	0.949	0.814	0.908
YOLOv7W6	70.4M	1280	0.89	0.873	0.903	0.950	0.960	0.822	0.911
YOLOv7E6	97.2M	1280	0.87	0.875	0.866	0.940	0.905	0.864	0.904
YOLOv7D6	154.7M	1280	0.86	0.856	0.869	0.835	0.958	0.790	0.895
YOLOv7E6E	151.7M	1280	0.90	0.903	0.888	0.951	0.949	0.827	0.909

V. CONCLUSIONS AND FUTURE WORK

This study has presented an algorithm to detect and identify good, broken, or missing cashews in the packaging line and check product quality. The CASHEW dataset was developed to test the model's effectiveness by collecting cashew images in different lighting conditions and camera angles. The YOLOv7 convolutional neural network model with many different versions was employed. The evaluation results revealed that the model can recognize cashew nuts with an average accuracy (mAP) of about 90%. Besides, the YOLOv7tiny model has the least number of parameters (6.2M) but has many parameters with higher accuracy than that of some other YOLO ones. In this paper, we have only applied the YOLOv7 model without any improvement. Therefore, it is possible to improve the YOLOv7 model, test and compare it with some other neural network models, and integrate the proposed algorithm on embedded systems with low configurations for the actual cashew packaging and quality inspection lines.

ACKNOWLEDGMENT

This work was supported and financed by Hanoi University of Industry, Project Granted number 15-2021-RD/HĐ – ĐHCN.

REFERENCES

[1] N. N. Oliveira, C. G. Mothé, M. G. Mothé, and L. G. de Oliveira, "Cashew nut and cashew apple: a scientific and technological monitoring worldwide review," *Journal of Food Science and Technology*, vol. 57, no. 1, pp. 12-21, 2020.

[2] M. Aremu, I. Ogunlade, and A. Olonisakin, "Fatty acid and amino acid composition of protein concentrate from cashew nut (Anarcadium occidentale) grown in Nasarawa State, Nigeria," *Pakistan Journal of Nutrition*, vol. 6, no. 5, pp. 419-423, 2007.

[3] A. Berry and S. Sargent, "Cashew apple and nut (Anacardium occidentale L.)," in *postharvest biology and technology of tropical and subtropical fruits*: Elsevier, 2011, pp. 414-423e.

[4] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3," *Sensors*, vol. 20, no. 7, p. 2145, 2020.

[5] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and electronics in agriculture*, vol. 157, pp. 417-426, 2019.

[6] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved YOLOv5," *Remote Sensing*, vol. 13, no. 9, p. 1619, 2021.

[7] S. K. Behera, A. K. Rath, A. Mahapatra, and P. K. Sethy, "Identification, classification & grading of fruits using machine learning & computer intelligence: a review," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-11, 2020.

[8] S. Naik and B. Patel, "Machine vision based fruit classification and grading-a review," *International Journal of Computer Applications*, vol. 170, no. 9, pp. 22-34, 2017.

[9] C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques," *Mathematical Problems in Engineering*, vol. 2022, 2022.

[10] X. Jin, Y. Sun, J. Che, M. Bagavathiannan, J. Yu, and Y. Chen, "A Novel Deep Learning-Based Method for Detection of Weeds in Vegetables," *Pest Management Science*, 2022.

[11] M. K. R. Asif, M. A. Rahman, and M. H. Hena, "CNN based disease detection approach on potato leaves," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020: IEEE, pp. 428-432.

[12] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 3, pp. 243-257, 2021.

[13] C. Cervantes-Jilaja *et al.*, "Optimal Selection and Identification of Defects in Chestnuts Processing, through Computer Vision, Taking Advantage of its Inherent Characteristics," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019: IEEE, pp. 513-520.

[14] S. Parvathi and S. T. Selvi, "Detection of maturity stages of coconuts in complex background using Faster R-CNN model," *biosystems engineering*, vol. 202, pp. 119-132, 2021.

[15] A. Sivaranjani, S. Senthilrani, B. Ashokumar, and A. S. Murugan, "CashNet-15: an optimized cashew nut grading using deep CNN and data augmentation," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019: IEEE, pp. 1-5.

[16] N. V. Ganganagowder and P. Kamath, "Intelligent classification models for food products basis on morphological, colour and texture features," *Acta Agronómica*, vol. 66, no. 4, pp. 486-494, 2017.

[17] K. T. Islam, S. Wijewickrema, M. Pervez, and S. O'Leary, "An exploration of deep transfer learning for food image classification," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018: IEEE, pp. 1-5.

[18] Z. Huang *et al.*, "Fast location and segmentation of high-throughput damaged soybean seeds with invertible neural networks," *Journal of the Science of Food and Agriculture*, 2022.

[19] K. Hameed, D. Chai, and A. Rassau, "A comprehensive review of fruit and vegetable classification techniques," *Image and Vision Computing*, vol. 80, pp. 24-44, 2018.

[20] V. Narendra and K. Hareesha, "Cashew kernels classification using colour features," *International Journal of Machine Intelligence*, vol. 3, no. 2, pp. 52-57, 2011.

[21] V. Narendra and K. Hareesha, "Cashew kernels classification using texture features," *International Journal of Machine Intelligence*, vol. 3, no. 2, pp. 45-51, 2011.

[22] M. Aran, A. G. Nath, and A. Shyna, "Automated cashew kernel grading using machine vision," in *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, 2016: IEEE, pp. 1-5.

[23] S. Sunoj, C. Igathinathane, and S. Jenicka, "Cashews whole and splits classification using a novel machine vision approach," *Postharvest Biology and Technology,* vol. 138, pp. 19-30, 2018.

[24] S. K. Vidyarthi, S. K. Singh, R. Tiwari, H. W. Xiao, and R. Rai, "Classification of first quality fancy cashew kernels using four deep convolutional neural network models," *Journal of Food Process Engineering,* vol. 43, no. 12, p. e13552, 2020.

[25] 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.

[26] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016: Springer, pp. 21-37.

[27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934,* 2020.

[28] 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.

[29] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[30] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263-7271.

[31] 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.

[32] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks,* vol. 168, p. 107036, 2020.

[33] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017: IEEE, pp. 3626-3633.

[34] R. Gai, N. Chen, and H. Yuan, "A detection algorithm for cherry fruits based on the improved YOLO-v4 model," *Neural Computing and Applications,* pp. 1-12, 2021.

[35] A. Koirala, K. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'," *Precision Agriculture,* vol. 20, no. 6, pp. 1107-1135, 2019.

[36] H. Mirhaji, M. Soleymani, A. Asakereh, and S. A. Mehdizadeh, "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,* vol. 191, p. 106533, 2021.

[37] D. Wang and D. He, "Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning," *Biosystems Engineering,* vol. 210, pp. 271-281, 2021.

[38] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for Kiwifruit defects based on YOLOv5," *Electronics,* vol. 10, no. 14, p. 1711, 2021.

[39] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430,* 2021.

[40] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," *arXiv preprint arXiv:2105.04206,* 2021.

[41] D. Tzutalin, "LabelImg," *GitHub Repository,* vol. 6, 2015.