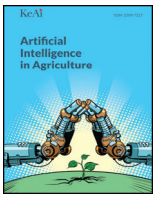
Artificial Intelligence in Agriculture 5 (2021) 102–117

Contents lists available at ScienceDirect

Artificial Intelligence in Agriculture

journal homepage: http://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/



Non-destructive thermal imaging for object detection via advanced deep learning for robotic inspection and harvesting of chili peppers

Steven C. Hespeler, Hamidreza Nemati, Ehsan Dehghan-Niri *

Intelligent Structures and Nondestructive Evaluation (ISNDE), Civil Engineering Department, New Mexico State University, Las Cruces, NM, USA

ARTICLE INFO

Article history:
Received 15 March 2021
Received in revised form 11 May 2021
Accepted 11 May 2021
Available online 15 May 2021

Keywords:
Deep learning
You only look once (YOLO) v3
Object detection
Chili pepper fruit

ABSTRACT

Deep Learning has been utilized in computer vision for object detection for almost a decade. Real-time object detection for robotic inspection and harvesting has gained interest during this time as a possible technique for high-quality machine assistance during agriculture applications.We utilize RGB and thermal images of chili peppers in an environment of various amounts of debris, pepper overlapping, and ambient lighting, train this dataset, and compare object detection methods. Results are presented from the real-time and less than real-time object detection models. Two advanced deep learning algorithms, Mask-Regional Convolutional Neural Networks (Mask-RCNN) and You Only Look Once version 3 (YOLOv3)are compared in terms of object detection accuracy and computational costs. When utilizing the YOLOv3 architecture, an overall training mean average precision (mAP) value of 1.0 is achieved. Most testing images from this model score within a range from 97 to 100% confidence levels in natural environment. It is shown that the YOLOv3 algorithm has superior capabilities to the Mask-RCNN with over 10 times the computational speed on the chili dataset. However, some of the RGB test images resulted in low classification scores when heavy debris is present in the image. A significant improvement in the real-time classification scores was observed when the thermal images were used, especially with heavy debris present. We found and report improved prediction scores with a thermal imagery dataset where YOLOv3 struggled on the RGB images. It was shown that mapping temperature differences between the pepper and plant/debris can provide significant features for object detection in real-time and can help improve accuracy of predictions with heavy debris, variant ambient lighting, and overlapping of peppers. In addition, successful thermal imaging for real-time robotic harvesting could allow the harvesting period to become more efficient and open up harvesting opportunity in low light situations.

1. Introduction

Due to its lightly pungent, crisp, and smoky taste, the New Mexico chili pepper (*Capsicum annuum*) is widely popular in the Southwest region of the USA. Sometimes referred to as chiles, the pepper in New Mexico is a cash crop with an annual harvesting of approximately 8000 to 10,000 acres and is used for consumption, processing into dried spice, or decorations (strung on ristras) (Bosland et al., 1991).

The agricultural production of chili peppers is the most consumed spicy crop throughout the world (Jiang et al., 2018). Throughout the world, the USA and other countries lack in total *Capsicum annuum* production compared to agricultural production achievements made by China in 2019. Fig. 1 displays the production quantities of the top 11 countries worldwide.

Total world production and area harvested have steadily increased each year since 1994. From 2009 to 2019, worldwide production of the crop increased by 10 million tonnes per year, from 28 to 38 million tonnes (FAO, 2019). Fig. 2 visualizes the worldwide trend. While demand for the chili pepper has increased, harvest production has room for growth. Agriculture producer prices (prices for crops at the initial sale point) do not indicate quantity values of production. Although one of the smaller total producing countries with the least amount of land harvested, Netherlands achieved the highest selling point per hectare (ha) of chili peppers among the entire group highlighted. Table 1 displays data from 2019 that showcases the production amount, land area harvested, producer price at the initial sale, and how much price each hectare produces for these countries (FAO, 2019). Using this data as a blueprint, countries with less land harvested could increase production to become more competitive and drive up their production yield per hectare. We aim to illuminate and provide insight into this issue.

Nondestructive imaging for robotic harvesting has been an effective tool for assistance in the agriculture industry to harvest the fruit with debris present while not harming the plant (Gao et al., 2020). Due to the complexity of this task, an efficient object detection and inspection algorithm are necessary for a robotic platform to be used in pepper

* Corresponding author.
E-mail address: niri@nmsu.edu (E. Dehghan-Niri).

https://doi.org/10.1016/j.aiia.2021.05.003

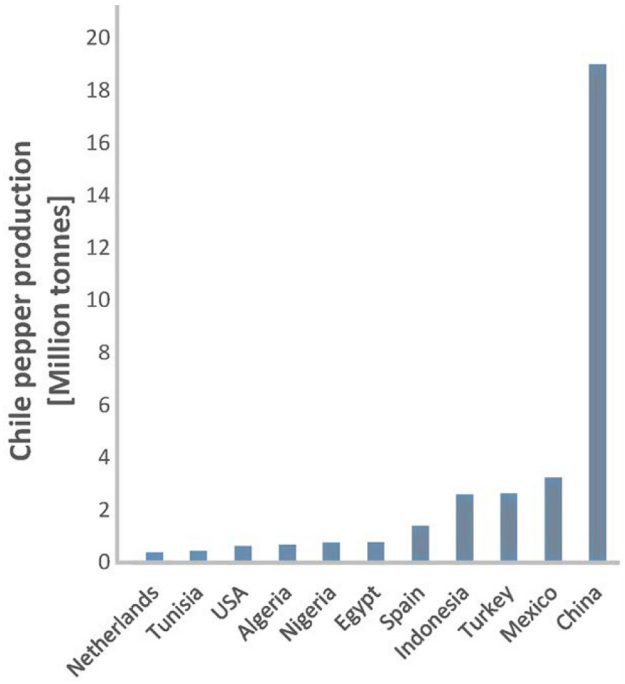


Fig. 1. Chili pepper production in 2019 (FAO, 2019).

harvesting. Nondestructive quality assessment of peppers has been studied successfully using near-infrared (NIR) hyper-spectral imaging to not harm the plant (Jiang et al., 2018).

Real-time robotic harvesting is a promising development in agriculture with some challenges (Kang et al., 2020). Challenges in this area result from complex machine tasks that require powerful computation abilities, complex model architectures, and non-destructive measures, all while the machine is detecting the fruit. With advances in Convolution Neural Network (CNN) techniques, object detection performance has rapidly increased in recent years and is a suitable solution for real-time robotic harvesting. Gu et al. showcase the recent growth of CNNs through a broad scope of literature (Gu et al., 2018). The survey highlights several improvements that have propelled the expansion of CNN applications. A critical development is applying several new activation functions that have increased in model performance. One extremely effective activation function is the Rectified Linear Unit (ReLU). The transition to ReLU from sigmoid function solved the famous issue of vanishing gradient, which was impeding traditional Neural Networks (NN) from learning on large datasets. ReLU can be mathematically represented as:

$$a_{i,j,k} = max\left(z_{i,j,k}, 0\right) \tag{1}$$

where $z_{i,j,k}$ is the input of the activation function located at (i,j). Other variants include Leaky ReLU (and many other variants), ELU, Maxout, and Probout. We refer the reader to the following reference for an in-depth analysis of CNN components (Gu et al., 2018).

Thermal imaging is used to convert invisible radiation patterns of a particular object to visible imaging for feature extraction (Vadivambal and Jayas, 2011). This type of imaging can be utilized as a non-destructive method during real-time robotic harvesting since it is non-contact, non-invasive, and rapid (Vadivambal and Jayas, 2011). We believe mapping temperature differences between the pepper and plant/debris can provide significant features for object detection in real-time and can help improve accuracy predictions with heavy debris, variant ambient lighting, and overlapping of peppers. In addition, successful thermal imaging for real-time robotic harvesting could allow the harvesting period to become more efficient and open up harvesting opportunity in the evening hours or low light situations. Fig. 3 highlights how prominent thermal imaging can be for extracting features than an RGB image.

Modern architectures (one-stage and two-stage) for object detection is implemented on a custom dataset to detect chili peppers in real-time for robotic harvesting. In this investigation, we 1) compare

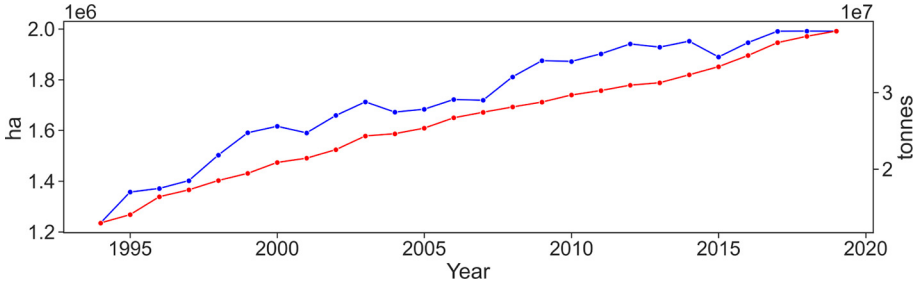


Fig. 2. Chili area harvested (blue) in ha and chili production (red) in tonnes (FAO, 2019).

Table 1
Chili pepper data 2019 (FAO, 2019).

Country	Production (tonnes)	Area Harvested (ha)	Producer Price (USD/t)	Price per ha
Netherlands	375,000	1500	1177	$ **294,150.00**
Tunisia	443,632	20,103	382	$ 8425.54
USA	624,982	19,627	941	$ 29,964.24
Algeria	675,168	21,767	643	$ 19,953.86
Nigeria	753,116	99,715	1077	$ 8130.47
Egypt	764,292	40,422	127	$ 2395.62
Spain	1,402,380	21,430	927	$ 60,662.91
Indonesia	2,588,633	300,377	1206	$ 10,388.93
Turkey	2,625,669	92,089	432	$ 12,308.76
Mexico	3,238,245	149,577	525	$ 11,370.24
China	**19,007,248**	**798,877**	748	$ 17,801.52

(a) (b)

Fig. 3. (a) RGB image, (b) Thermal image.

imagery features on RGB and thermal datasets, 2) compare two modern object detection techniques on an RGB dataset for locating and identifying chili fruit, and 3) develop a thermal imaging technique to improve object detection on images with high debris, various ambient lighting, and overlapping peppers using YOLOv3.

2. Literature review

2.1. Object detection

Object detection has a rich history of literature with vast improvements made recently that utilize computer vision to identify where and what an object is in space. Literature generally accepts the notation that two eras exist for the progression of object detection; traditional object detection before 2014 and the introduction of Deep Learning (DL) for object detection after 2014 (Zou et al., 2019). Before DL was utilized for object detection, many investigations were limited due to computation power and trainable image representation. We refer the reader to (Zou et al., 2019) for a comprehensive survey on object detection.

2.1.1. Traditional object detection

Viola and Jones (2001), introduced the Viola-Jones detector with the purpose of robust rapid and real-time detection. In this work, the authors focus on simple features rather than pixels to speed the detection process up. Three main contributions were made in the study. 1) Integral Image which is a method for using intermediate image representation for rapid rectangle computation, 2) an algorithm based on AdaBoost for feature extraction by combing some features to form an effective classifier, and 3) a cascade method for classification to discard or pass on viable information for further analysis and ultimately, increased detection performance. Viola and Jones (2004) demonstrated the robustness of this method by applying the techniques to facial recognition. Viola et al. (2005), utilized the cascade method for moving person detection along with AdaBoost for increasingly complex rejection regions based on a rule system.

Another cornerstone in object detection was introducing the Histograms of Oriented Gradient (HOG) detection system presented in Dalal and Triggs (2005). In this investigation, Dalal et al. created a method that utilizes edge direction and gradient intensity to determine the appearance of an image. This task is accomplished by separating the window into cells (spatial regions) and combining localized histograms of gradients directions and edge orientations over every pixel. After the HOG descriptors are calculated, they are fed to a linear Support Vector Machine (SVM) for classification purposes.

The Deformable Part-based Model (DPM) was developed by Felzenszwalb et al. (2008) and builds off of the architecture from (Dalal and Triggs, 2005). Using the underlying blocks, the model establishes HOG descriptors via histogram gradient magnitudes within each 1D pixel. Filters are used to dictate weights in the detection window. Learning is accomplished on the PASCAL training data with a latent SVM. Using DPM, the authors won 2007, 2008, and 2009 Pascal Visual Object Classes (VOC) detection challenge (Everingham et al., 2007).

2.1.2. Deep learning for object detection

For a comprehensive survey of Convolutional Neural Networks (CNN), we refer the reader to (Gu et al., 2018). DL approaches advanced significantly in 2012 with the introduction of AlexNet that was based on LeNet-5 architecture, just with a deeper structure (Krizhevsky et al., 2012).

Two -step-based detectors examine the input (image), apply regional proposals, and detect objects. Then, those detected objects are cropped and processed by an entirely separate network to estimate. Two-step methods typically require more computation time because each step requires resampling (three in total) (Poirson et al., 2016). Typically, one-stage detector networks divide regions and apply prediction bounding boxes with probabilities inside each region simultaneously. Fig. 4 illustrates the two-stage (4a) and one-stage process (4b).

2.1.3. CNN two-stage based detectors

Girshick et al. (2014) introduced the Regional CNN (R-CNN). They reported drastic improvements (at the time) to mean precision average (mPA) metrics compared to the best performing method on VOC2012. It begins with extracting around 2000 bottom-up region proposals (object candidate boxes) by Selective Search (Uijlings et al., 2013). Next, each regional proposal was rescaled to fixed image size (227 × 227) and used for model training via a CNN on an image database like ImageNet (Deng et al., 2009) to extract features. Lastly, a binary SVM classifier is used to predict/detect an object(s) within each region.

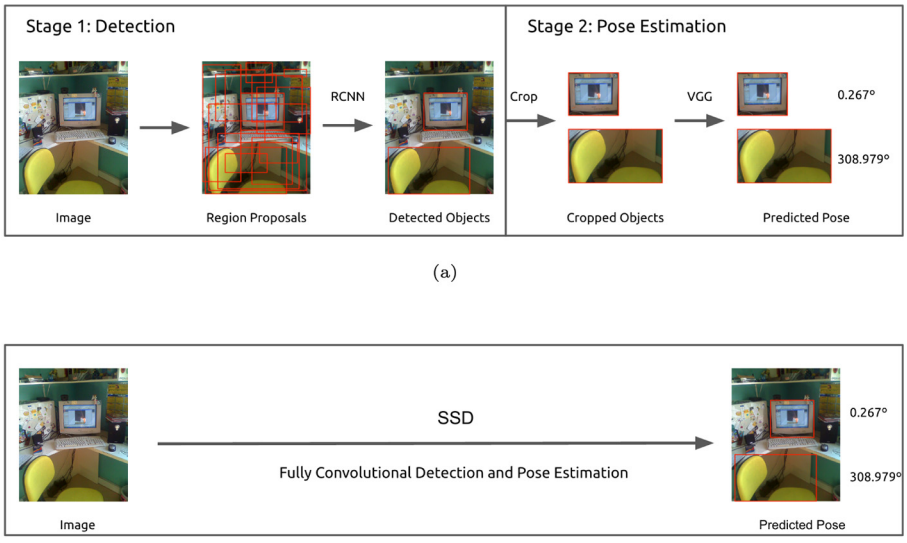


Fig. 4. Two/One-stage detectors (Poirson et al., 2016): (a) Two-stage, (b) One-stage.

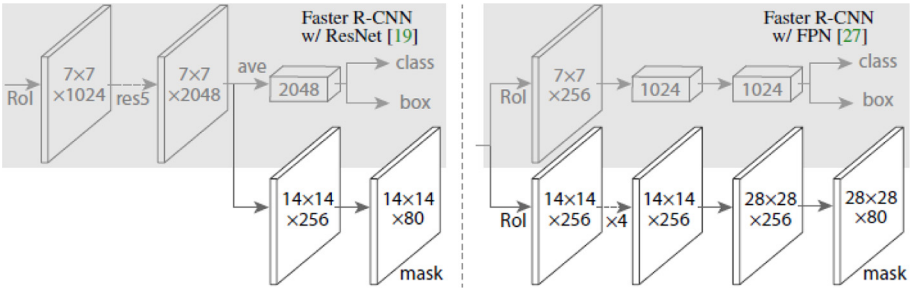


Fig. 5. Head architecture for Mask R-CNN (He et al., 2017).

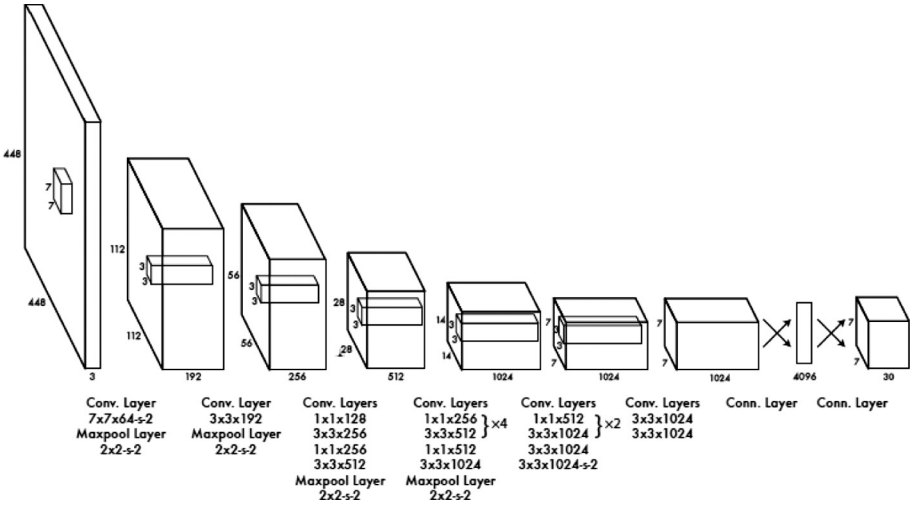


Fig. 6. YOLO architecture (Redmon et al., 2016.

Girshick (2015), improved on the R-CNN by simplifying the learning process. The authors realized that the R-CNN is slow due to the network performing a "forward pass for each object proposal". Spatial Pyramid Pooling networks (SPP or SPPnet) were introduced by He et al. (2015) in an attempt to speed up training. Gu et al. (2018), utilized SPP to update all network layers during fine-tuning. Ren et al. (2015), made

Table 2
YOLO performance (Redmon et al., 2016).

Real-Time Detectors	Train	mAP	FPS
100 Hz DPM (Sadeghi and Forsyth, 2014)	2007	16.0	100
30 Hz DPM (Sadeghi and Forsyth, 2014)	2007	26.1	30
Fast YOLO	2007 + 2012	52.7	***155***
YOLO	2007 + 2012	***63.4***	45
Less Than Real-Time			
Fastest DPM (Yan et al., 2014)	2007	30.4	15
R-CNN Minus R (Lenc and Vedaldi, 2015)	2007	53.5	6
Fast R-CNN (Girshick, 2015)	2007 + 2012	70.0	0.5
Faster R-CNN VGG-16 (Ren et al., 2015)	2007 + 2012	73.2	7
Faster R-CNN ZF (Ren et al., 2015)	2007 + 2012	62.1	18
YOLO VGG-16	2007 + 2012	66.4	21

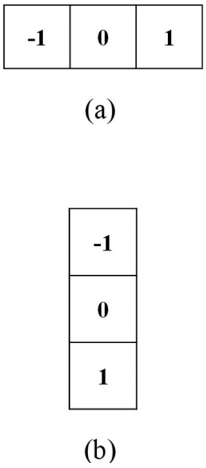


Fig. 7. HOG filter: (a) Horizontal kernel, (b) Vertical kernel.

improvements on the fast R-CNN design in the same year to move the algorithm closer to real-time detection speeds. This version became the first end-to-end and close to real-time object detector (Zou et al., 2019). In the first stage, a Region Proposal Network (RPN) considers a candidate bounding box. In the second stage, feature extraction is completed using Region of Interest (RoI) pooling (RoIPool) from each candidate which executes classification and bounding box regression (Ren et al., 2015).

Mask R-CNN is an extension of the fast R-CNN architecture and generates a high-quality segmentation mask for each instance while efficiently detecting objects in an image (He et al., 2017). Using various Feature Pyramid Network (FPN) frameworks, R-50-FPN, R-101-FPN, X-101-FPN, and X-152-FPN, AP performance was measured at 40.8 and 70.4 for enhanced detection and enhanced keypoint results, respectively. Fig. 5 details the head architecture of the algorithm and highlights the exact improvements made to the Faster R-CNN.

Here, we see the algorithm applied with the ResNet (left) and FPN backbones.

2.1.4. CNN one-stage based detectors

You Only Look Once (YOLO) is a real-time detecting system created by Redmon et al. (2016) with several incremental improvements through the years (Redmon and Farhadi, 2017; Redmon and Farhadi, 2018). Most recent improvements have been made by Bochkovskiy et al. (2020). The original system design gets its name from only looking at an image once, therefore treating every image (input) as a regression problem. Compared to R-CNN architectures that are complex pipelines, this improvement in simplicity might be the most important aspect of the YOLO system. The system consists of 3 main steps. First, inputs are resized to 448 × 448. Next, a single CNN is run. Finally, the Non-max Suppression (NMS) technique attempts to correct multiple detections on the same image.

The YOLO architecture consists of 24 convolutional layers followed by 2 fully connected layers. Fig. 6 shows the details of each layer in the basic YOLO architecture. The network outputs a 7 × 7 × 30 tensor of predictions.

In 2018 Redmon et al. established an incremental improvement to the network design to include 53 convolutional layers. This version demonstrated exceptional accuracy due to the more robust design and is 3.8 times faster than RetinaNet. At the time of publication (2016), the YOLO system achieved more than twice the mPA compared to the next best real-time system, 63.4% mAP to 26.1%, respectively. It also observes the entire image, unlike region proposal-based architectures. Table 2 summarizes the performance of YOLO compared to other real-time and less than real-time systems (Redmon et al., 2016).

Another critical aspect of YOLO performance is that when compared to Fast R-CNN, YOLO had a smaller percentage of background errors but more significant percentage of localization errors. DL, especially CNNs, have been thrust upon the robotic harvesting research area for assistance during agriculture applications. The research area has seen a surge in investigations in 2019 and 2020 (Naranjo-Torres et al., 2020). In Hameed et al. (2018), investigators present a critical comparison of multiple modern computer vision techniques investigated by researchers to classify fruit and vegetables. In the study, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Decision Trees, Artificial Neural Networks (ANN), and CNN's are reviewed for a variety of fruit and vegetable classification. However, the Mask R-CNN and YOLOv3 methods are not evaluated in the investigation.

Li et al. (2019), reviewed non-destructive optical techniques for berries, mainly strawberries and blueberries. The researchers identified and reviewed 13 methods, one of which was computer vision. The review did not detail specific computer vision techniques; however, several studies are mentioned that detail computer vision as a viable method for berry classification. The researchers state that one main issue with computer vision is “inconsistent ambient illumination and complex background.” For these reasons, we have selected the YOLOv3 for real-time object detection.

2.2. Object detection in agriculture

In the past decade, different object detection methods have been successfully adopted in several agricultural applications. Hameed et al.

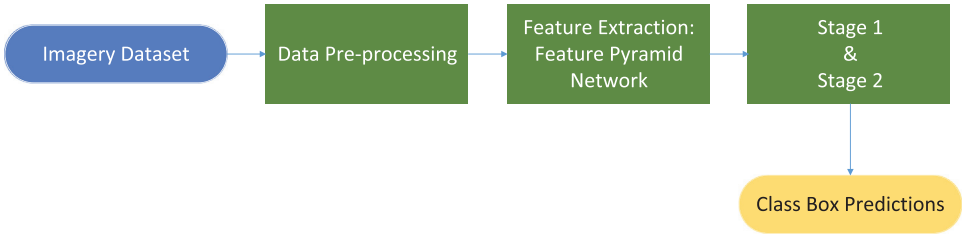


Fig. 8. Mask R-CNN block diagram.

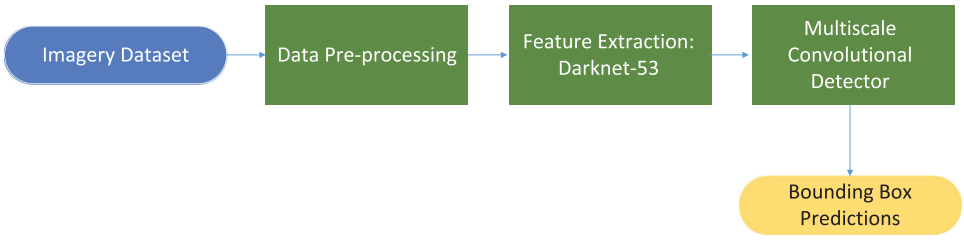


Fig. 9. YOLOv3 block diagram.

Fig. 10. RGB HOG images: (a) Original RGB image, (b) Resized RGB image, (c) HOG RGB.

highlighted the complexities of classifying fruit and vegetables. Specifically, they looked at the issue of identifying fruits and vegetables at supermarket self-checkout systems (Hameed et al., 2020a). The authors applied pre-trained weights via transfer learning and an ensemble method to weights of GoogleNet and MobileNet for an optimized average weight. Results demonstrated that these pre-trained weights positively affected the model, with the ensemble method reaching accuracy measurements higher than training and testing datasets. Hameed et al. (2020b), proposed a novel technique to the supermarket fruit and vegetable classification issue by incorporating a progressive fruit and vegetable classifier with AdaBoost-based optimization of a CNN. 15 classes, 1000 images in each class, of fruits and vegetables were considered for training. The method began by coarsely classifying fruits and vegetables into three categories via the Jenks Natural Breaks classification method. From there, the classes were implemented to three CNNs (GoogleNet, MobileNet-v2, and a custom-designed network) for more detailed classification. The optimized AdaBoost CNN outperformed the custom CNN. Accuracy of the CNN with AdaBoost optimization ranged from 97.60% (navel orange) to 99.87% (ladyfinger banana) with less than 3% error for all classes. The custom 15 layer CNN achieved test accuracy ranges of 80.13% (10 epochs) to 93.97% (22 epochs).

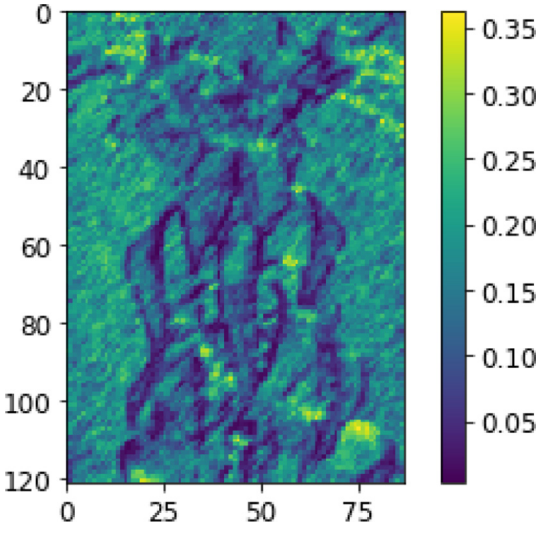


Fig. 11. Gradients of images.

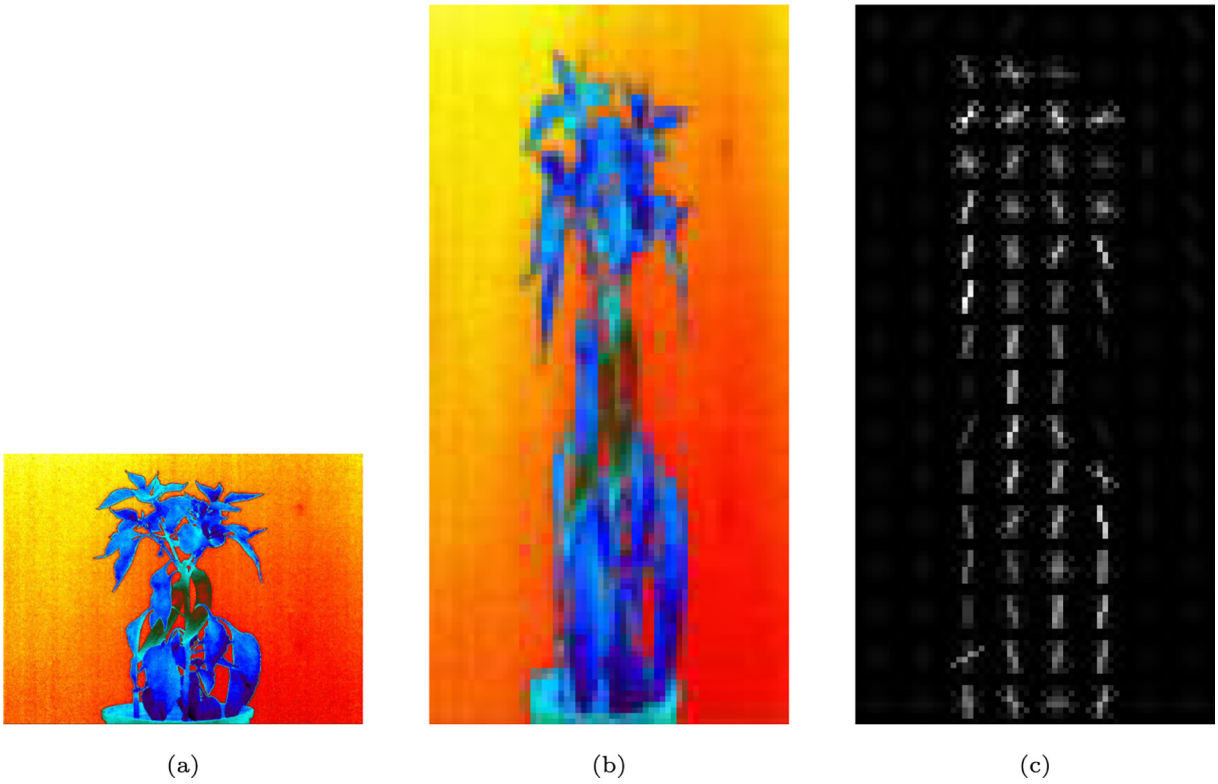


Fig. 12. Thermal HOG images: (a) Original thermal image, (b) Resized thermal image, (c) HOG thermal.

Several recent studies have attempted to improve or test modern object detection methods to use real-time robotic harvesting. Wan and Goudos (2020), utilized an improved Faster R-CNN architecture to detect fruits (apples, mango's, oranges) for robot picking applications. In the investigation, the authors reported on 1) how their model updates hyperparameters during the training phase of their model, 2) how the model augments data on high-quality images, and 3) optimization of the convolutional and pooling layers of their Faster R-CNN model. Specifically, feature extraction was completed with a CNN model VGG-16 using 13 convolutional layers, 13 ReLu layers, and four pooling layers. Two loss functions were added to optimize the convolutional and pooling layers, allowing the parameters to adjust automatically during training. They found that automatically tuning parameters resulted in 58 ms/image detection speed with mAP % of 90.72. These results were compared to YOLO, YOLOv2, YOLOv3, Fast R-CNN, and Faster R-CNN, of which their model outperformed YOLO, Fast R-CNN, and Faster R-CNN in terms of detection speed and mAP%. When the authors fed their dataset to YOLOv3 they achieved a 40 ms/image detection speed and mAP % of 90.03. So, they were able to soften the gap of performance between the Faster R-CNN and YOLOv3 architectures.

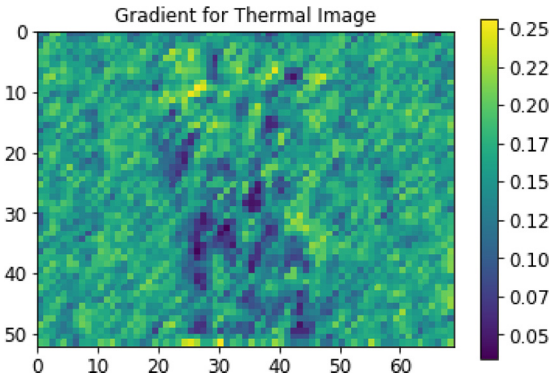


Fig. 13. Gradients of thermal images.

There have been studies that focus on low harvesting rates and how to improve picking point accuracy in real-time (Yu et al., 2020). This investigation, completed by Yu et al., involved creating an end-effector assembled on a servo control system of a robotic harvesting machine. This mechanism effectively allowed the robot to emit and receive a laser beam instead of measuring the depth distance in real-time. A rotated YOLO (R-YOLO) was proposed to improve to the YOLOv3 architecture to be compatible with the mechanism. A highlight of R-YOLO was the process of rotating the annotation bounding boxes and employing a rotational parameter, α, during the labeling phase. This process aids in the localization of the picking point of the fruit. They reported precision and recall rates of 94.4 and 93.4%, respectively. The average computational speed on a 640 × 480 image with R-YOLO was 0.056 s.

Robust and efficient non-destructive testing methods play an essential role in robotic harvesting and quality control of agricultural products. Throughout literature, many novel techniques have been implemented for accomplishing this task. Jiang et al. (2018), employed near-infrared hyperspectral imaging for non-destructive quality assessment of chili peppers. In combination with regression models, NIR was used for evaluating capsaicinoid concentrations and the water content of peppers. Besides, a radial basis function neural network (RBFNN) was used for classifying pungent and non-pungent peppers. The investigation demonstrated that capsaicin and dihydrocapsaicin concentrations of chili peppers are visually different. Pungency was classified with an accuracy of 98.7 and 98.0% (full spectra and successive projections algorithm, respectfully). Gao et al. (2020), implemented a real-time hyperspectral imaging for fieldwork in classifying the ripeness of fruit. This investigation reported a technique created by the authors

that utilized a portable hyperspectral imagery machine to estimate the ripeness of strawberries. Two wavelengths were selected based on a forward SFS algorithm and feed to an SVM. Finally, a CNN was implemented to predict deep spatial features, and the model achieved an accuracy of 98.6%.

3. Methodology

3.1. Image representation

Vision-based fruit detection is a critical component for in-field automation and robotic fruit harvesting. However, this technology could also be leveraged in other applications such as disease detection, maturity detection, crop health status monitoring (Patel et al., 2011).

The datasets in this study were constructed as thermal and RGB color images to detect the chili peppers. To this end, we monitored two different chili plants during their growth period. The first chili pepper was a raw, green one, and the second was mature. The RGB images were acquired from conventional digital cameras with no preprocessing and saved on a standard JPEG file format. At the same time, thermal images of chili peppers were collected using a Forward Looking Infrared camera (FLIR A615). The camera is highly accurate with easy setup and automated inspection software. It is sensitive enough to detect temperature changes as small as 50 mK. The spectral range of the thermal camera is $7.5 - 14\ \mu$m with an absolute thermal accuracy of ± 2°C or ± 2%, and thermal sensitivity of <0.05° at 30 °C. The camera produces thermal images with up to 640 by 480 resolution at 60 frames per second.

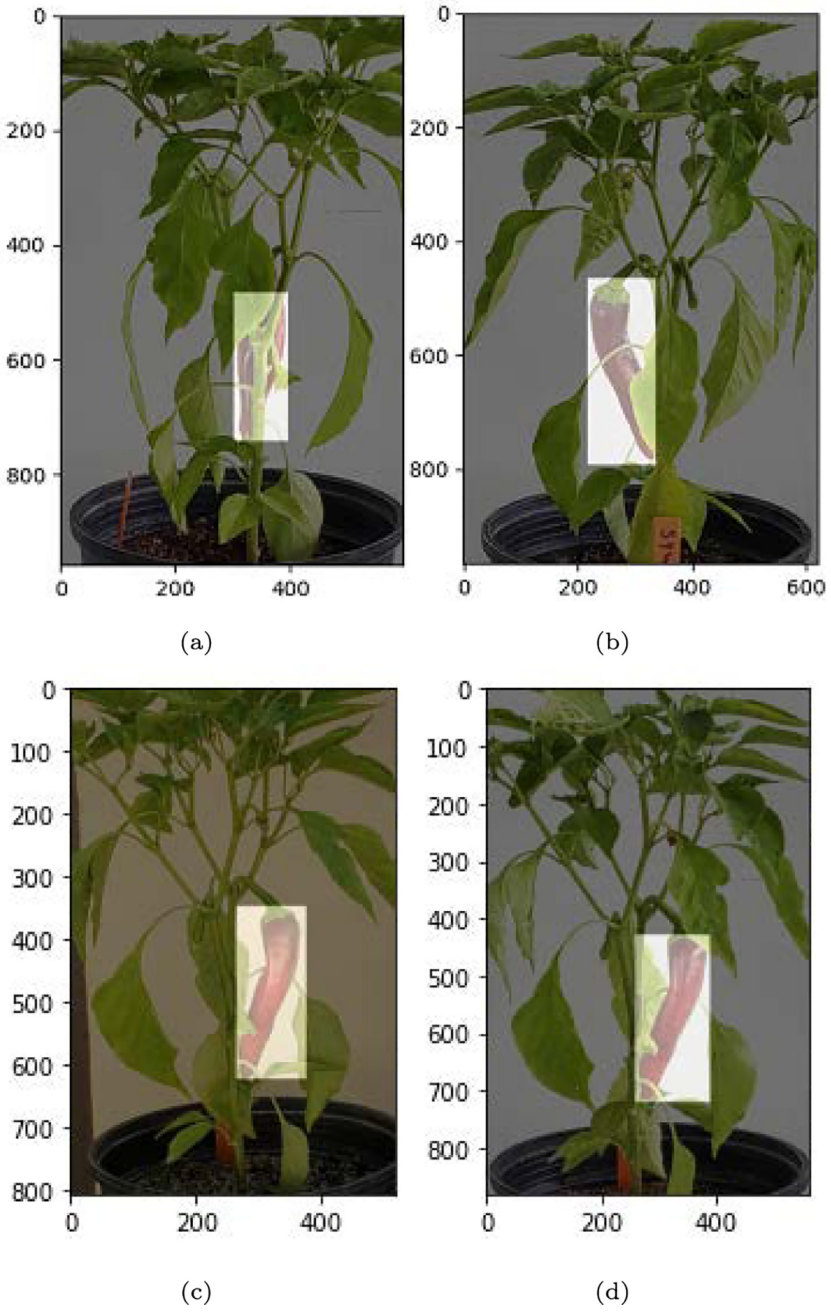


Fig. 14. Images with masks: (a)Mask 1, (b)Mask 2, (c) Mask 3, (d) Mask 4.

Both cameras were mounted over the chili plants at a fixed distance of 130 cm with a light background. The thermal camera was fully controlled with the FLIR Tools; In fact, pre or post-processing could be done on an individual image, using ResearchIR software. Using an IR lens with a focal length of 13.1 mm, 112 thermal images (Besides 112 RGBs) were captured every other day over three months. On each day, four different images were taken from different directions for a single plant. Some of the images were captured in an outdoor setting and in natural light to validate the proposed technique.

3.2. Performance metrics

Unlike other ML applications, object detection classification is not binary, as in true or false. Therefore, the performance metrics used have been tailored for these exact operations. Intersection of Union (IoU) gives the user the ability to measure how well bounding boxes are being predicted with respect to truth boxes greater than a user-identified threshold. In other words, IoU is a measurement of the object localization accuracy between the truth boxes and predicted boxes (Zou et al., 2019). Eq. 2 defines IoU:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2}$$

Here, we are measuring the magnitude of overlap between the ground truth box and predicted box. In literature, an accepted threshold tends to be 0.5. That is:

$$\begin{cases} IoU \geq 0.5 & TP \\ IoU < 0.5 & FP \end{cases}$$

So, if the measurement is at or higher than 0.5 the object will be successfully identified. Precision and recall are used as a metric to evaluate performance. We define precision and recall in Eqs. 3 and 4 below.

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

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

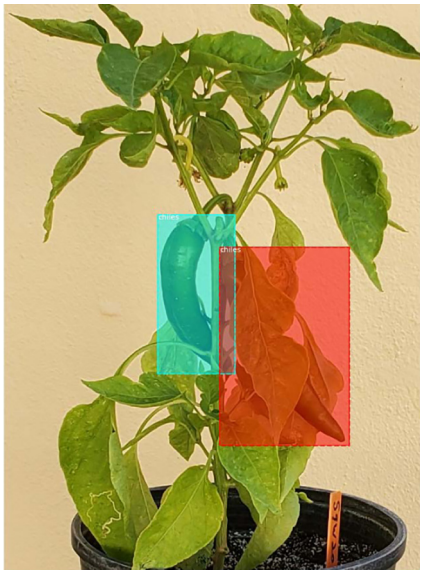


Fig. 15. Mask R-CNN prediction box.

Where, TP is true positive, FP is false positive, and FN is false negative. True positives are when the model correctly classifies the object detection. False positive occurs when there is incorrect object detection. False negative occurs when ground truth bounding boxes are present in the image and the model fails to detect the object in the image. True negative (TN) is not considered for this application since it would be correctly not labeling parts of the image that do not have truth bounding boxes.

3.3. Implementation of object detection models

Annotation files were generated for each image with the LabelImg software (tzutalin, 2017). Mask R-CNN was implemented in a Conda environment and operated with Python 3.6, Tensorflow 1.14, and Keras 2.24 packages. Next, we cloned the Github repository Mask R-CNN from (Abdulla, 2017).

The YOLOv3 model was computed on GPU with CUDA version 10.1, V10.1.243. The architecture was cloned from the Github repository created by Bochkovskiy (2020).

3.3.1. HOG implementation in Python

A prerequisite for using the HOG method is that all images must be an array of shape (128,64). Therefore, all images were dropped then transformed using skimage resize tool in python (van der Walt et al., 2014). The HOG descriptors work by applying a filter to the image. The vertical and horizontal gradients are calculated with the kernels shown in Fig. 7.

Magnitude and direction are calculated by Eqs. 5 and 6, respectively:

$$g = \sqrt{g_x^2 + g_y^2} \tag{5}$$

$$\theta = arctan\ \frac{g_y}{g_x} \tag{6}$$

3.3.2. Implementation of one and two-stage detection models

We select two top-performing algorithms from two-stage and one-stage detectors for implementation on our dataset. After the data was prepared, we split our test/train sets to 20/80% to train on as many images as possible due to the dataset being small. As stated in Section 2.1.2, the Mask R-CNN utilizes FPN and a ResNet101 backbone. For the Mask R-CNN model, we first loaded bounding boxes and then masks for each image. We utilized a python tool to ensure that the masks were applied correctly to the dataset. Then, we defined and fit the model using the MaskRCNN function supplied from Abdulla (2017). Fig. 8 depicts how our image dataset walks through the Mask R-CNN.

For our implementation of the Mask R-CNN architecture, we utilize FPN for feature extraction which is leveraged from a ConvNet's pyramidal feature hierarchical system (Lin et al., 2017). To do this, a region-based object detector or Region-of-Interest (RoI) pooling extracts features from the matrix. The RoI is assigned as:

$$k = \lfloor K_0 + log_2\left(\sqrt{wh}/224\right) \rfloor$$

where w and h are the width and height (respectively), on the input image to the network and K_0 is the target level. During the first stage, regions are proposed with anchors. Then the object class is predicted in the second stage. During the second stage, anchors are not used. Instead, ROIAlign is performed to fix misalignments (He et al., 2017). The ROIAlign generates masks for each object at the pixel level while the RoI classification branch is used to predict categories (He et al., 2017). Results are displayed in Section 4.2.

YOLOv3 consists of 53 convolutional layers that contain a batch normalization layer with a ReLU activation function. Pooling is not utilized for the model. Since this model is invariant to image size, cropped

images were similar in size but not the same. Feature maps are downsampled via a convolutional layer with a stride of 2. Fig. 9 walks through the process of how our data is implemented into YOLOv3. Once the dataset is prepared, the YOLOv3 model extracts features with Darknet-53. During feature extraction, three feature matrices are created of sizes 1) 52×52, 2) 26×26, and 3) 13×13. These feature matrices are fed to the multiscale convolutional detector where the features are concatenated through multi-step convolutions.

We batch and input images with size $(m,416,416,3)$ and output bounding boxes labeled $(p_c, b_x, b_y, b_h, b_w, c)$. The bounding box predictions are calculated by:

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

where, b_x, b_y, b_h, *and* b_w are bounding box center coordinates at x, y and width and height of the prediction box. $t_{x,\ y,\ w,\ h}$ are network outputs. Object scores are calculated as the probability (between 0 and 1) that an object is inside a bounding box from a sigmoid activation function. We report all findings in Section 4.3.

Fig. 16. Actual vs predicted plots for Mask-RCNN: (a) Actual plot image 1, (b) Predicted plot image 1, (c) Actual plot image 2, (d) Predicted plot image 2.

4. Results and discussion

This section provides the results of all experimentation completed during training. Here we show the viability of certain techniques applied to a chili pepper dataset in an environment with complex backgrounds and good ambient illumination. Section 4.1 details the HOG algorithm results, Section 4.2 provides results for the Mask R-CNN architecture, and Section 4.3 shows results from real-time detection using the YOLOv3 architecture. The RGB dataset was considered for experimentation on the HOG, Mask R-CNN, and YOLOv3 models.

Computer vision is a highly researched area with many new and cutting-edge technology. As a result, terminology can sometimes get confusing. AP was first introduced by VOC2007 (Everingham et al., 2007) and the calculation is applied to every image within the labeled set and measures the average AP of the entire set. We follow the COCO definition for mAP terminology as:

AP is averaged over all categories. Traditionally, this is called "mean average precision" (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context (Common Objects in Context (COCO) (2021)).

Therefore, we define mAP as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{7}$$

where, AP_i is the average precision at image i and N is the total number of images. The mAP metric is usually used as a final metric to compare performance throughout all object categories (Zou et al., 2019), which we use to evaluate experiments with both the Mask R-CNN and YOLOv3 models.

4.1. HOG results

The HOG method allows for visual feature extraction and comparison, which can be seen in Figs. 10, 11, 12, and 13. The purpose of using the HOG method in this experiment was to visualize gradient direction and intensity between RGB and thermal images of chili peppers. In Fig. 10, we show the process of transforming the original image in Fig. 10a, resizing the image in Fig. 10b, and finally plotting gradient directions in Fig. 10c. When plotting the normalized histograms in Fig. 10c, there is visual evidence (intensity) of the HOG detector recognizing the shape of the entire plant and even demonstrating a connection to the chili pepper centralized in the original image.

Gradient intensity of the histograms is visually represented as they increase in color in the image from the color map in Fig. 11. Gradient values range from 0 to 0.35 and visualize distinct features of the chili plant and pepper. Edges of the pepper and leaves have lower gradient values when compared to the background of the image. These values are close to zero (purple), while the center of the pepper display gradients between 0.30 and 0.25. The background displays gradient values ranging between 0.20 and 0.35. Here we show the regional location of where the chili pepper is within the normalized histogram plot. A plot of the gradients demonstrates evidence that the HOG method can outline the chili pepper in Fig. 11.

When we move to a thermal image, HOG detectors appear to have less visual evidence of detecting a pepper. Fig. 12 depicts features of a thermal image. The feature extraction process is represented with the original image in Fig. 12a, the resized thermal image in Fig. 12b, and the gradient directions in Fig. 12c. Gradient directions from the 12c plot does not show the distinct region of the pepper, but it does demonstrate the shape of the entire plant.

Gradients range from 0 to 0.25 in Fig. 13 and display the gradient values of the thermal image from Fig. 12a. The plot of gradient values does not demonstrate a clear feature pattern and the variation between gradient values is much less in Fig. 13 compared to Fig. 11 (<0.1). The HOG algorithm was an effective technique to visualize how features may be represented in computer vision. Figs. 11 and 13 demonstrate through gradient intensities that RGB images have stronger features when compared to thermal images.

4.2. Mask RCNN

Due to its ability to produce highly accurate mAP responses, we utilized the Mask R-CNN architecture to analyze and compare results from the Mask R-CNN to the YOLOv3 model. Before training, we ensure that image and mask arrays have identical width and height. Fig. 14 shows four images with masks overlaid to confirm masks were loading

Fig. 17. Predictions from pre-trained YOLO Weights: (a) Out-of-Box chili pepper prediction, (b) TRAINED chili pepper prediction.

properly. For example, Fig. 14d output an image shape of (881,562,3) and a mask shape of (881,562,1). We utilize the following tool:

visualize.display_instance

This python tool is used to call the plot in Fig. 15. Here, we demonstrate the ability of this particular architecture to detect chili peppers in an environment with debris. We visualize that the Mask R-CNN algorithm has identified and located all chili peppers in the image (with different colors), assigned a prediction bounding box, and labeled it with the appropriate label (chili pepper). As covered in Section 3.2, precision and recall are determined by evaluating truth bounding boxes and predicted bounding boxes. While training on the RGB dataset, the Mask R-CNN performed with a train mAP of 0.872 and test mAP of 0.896 with an overall computational time of 40.79 s per test image.

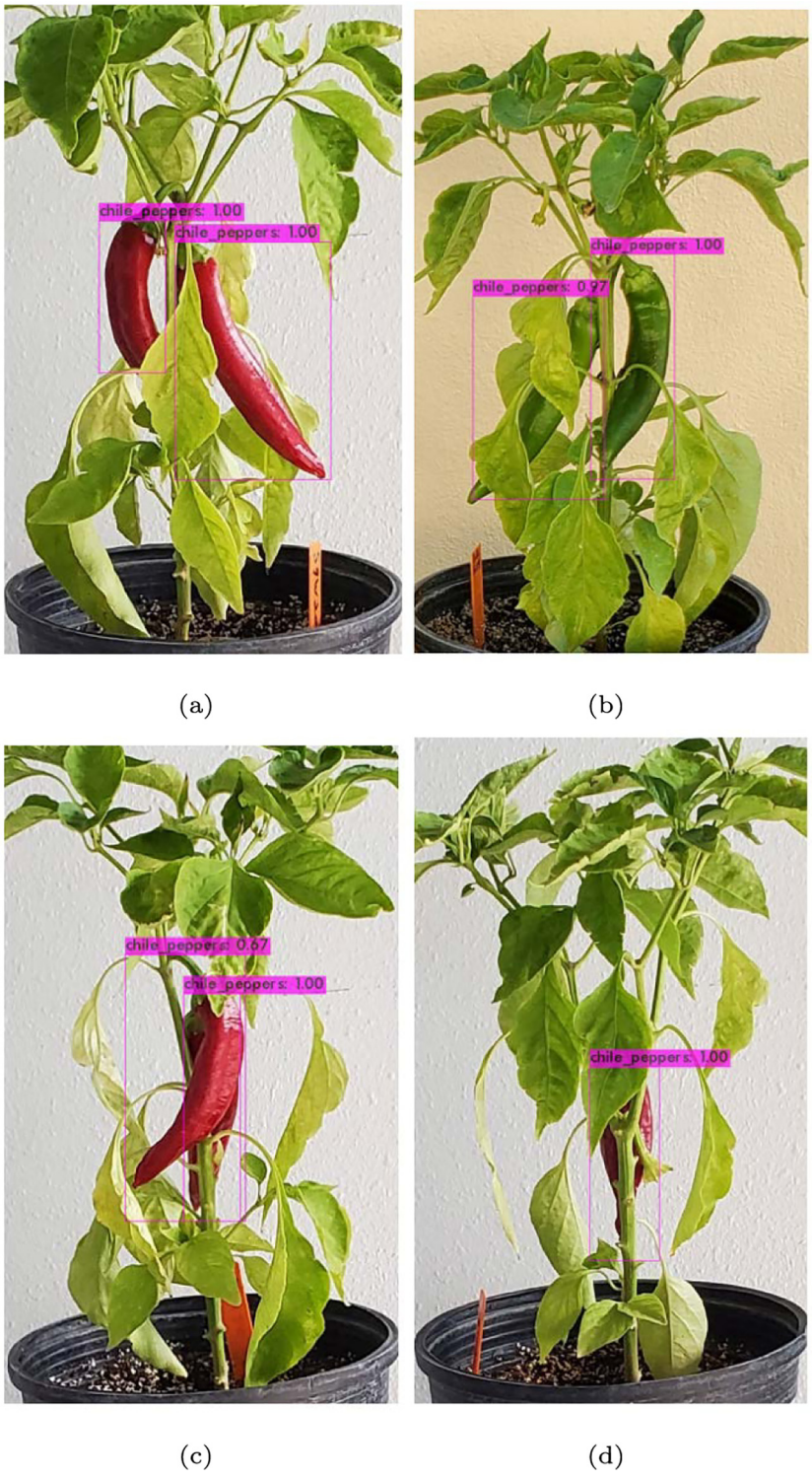


Fig. 18. Additional predictions on validation dataset with YOLOv3: (a) Prediction test image 2, (b) Prediction Test Image 3, (c) Prediction Test Image 4, (d) Prediction test image 5.

Fig. 19. Predictions on green chili test set via YOLOv3: (a) Prediction test image 2, (b) Prediction test image 3, (c) Prediction test image 4.

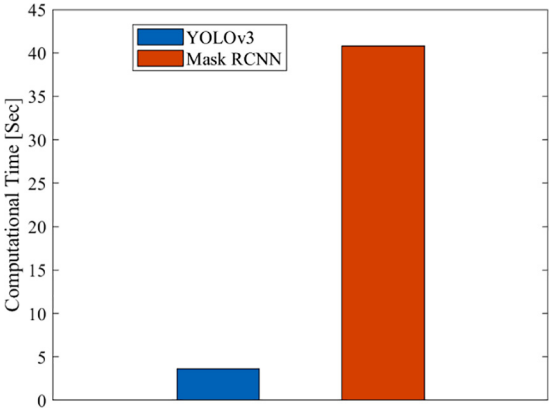


Fig. 20. Computational time (sec) for two-stage and one-stage algorithms.

Fig. 16 visualizes the truth and prediction bounding boxes for 2 images of chili peppers during testing. Overall, the Mask R-CNN model appears to place predicting boxes well. Fig. 16b, shows that the model failed to detect (False Negative) a chili pepper covered in debris.

4.3. YOLOv3 for real-time detection

We utilize the YOLOv3 architecture for real-time application due to its high performance and fast computing capabilities. We noticed that

Table 3
Mask R-CNN mAP values.

RGB		Thermal	
Train	Test	Train	Test
0.872	0.896	0.391	0.298

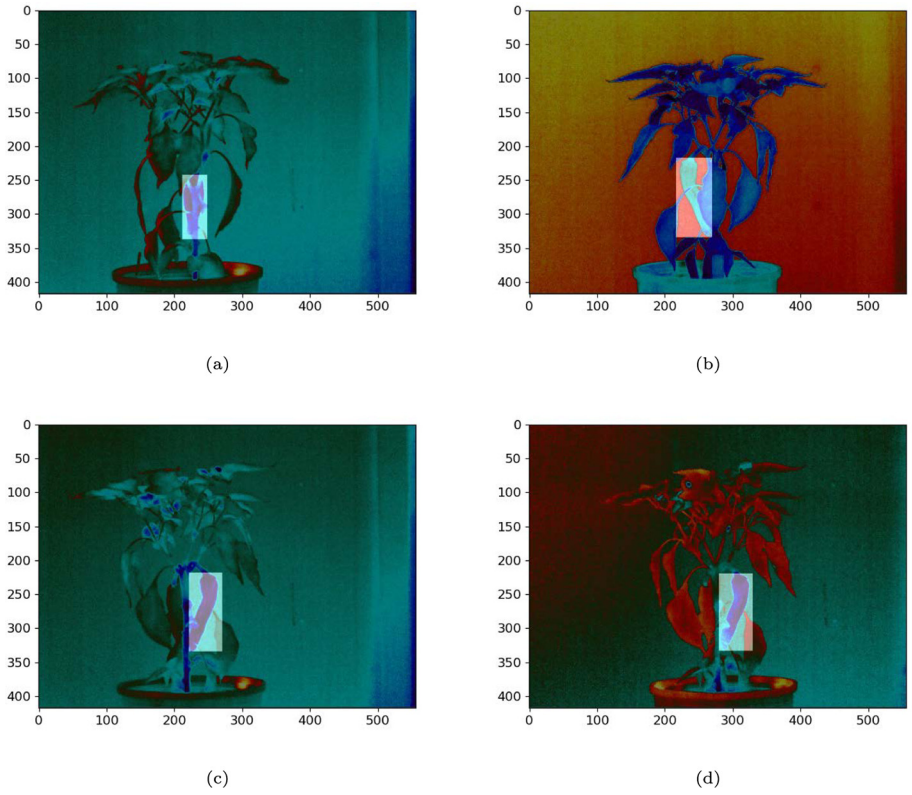


Fig. 21. Thermal images with Masks: (a)Mask 1, (b)Mask 2, (c) Mask 3, (d) Mask 4.

training loss drops below 2.0 roughly at the 250th epoch. After training was complete, we evaluated the model using mAP performance metrics. Both training and testing resulted in 100% precision accuracy (mAP) with a computational time of 3.64 s per test image.

An initial test was conducted using out-of-box pre-trained weights and classes on the chili dataset. Fig. 17 compares the out-of-box likeness scores of a chili pepper to that of a chili pepper image after training was complete.

Fig. 18 displays confidence levels of additional test images. As stated earlier in the text, confidence scores mostly evaluate test images between 97 and 100% as the predefined class of 'chile pepper'.

Confidence scores suggest that the YOLOv3 model is functioning with a high level of precision. There does appear to be an issue of debris in Fig. 19 a). Here, we can see that one of the peppers is only detected with a confidence of 33% which indicates a possible impediment during real-time robotic harvesting where the environment may be dense with debris.

Fig. 20 compares the computational capabilities of Mask-RCNN (red) to YOLOv3 (blue). The YOLOv3 algorithm has superior capabilities to the Mask-RCNN with over ten times the computational speed on the chili dataset. For this reason, we believe the YOLOv3 algorithm will be suitable in a harvesting robot for real-time object detection. Computational speeds for the YOLOv3 and Mask R-CNN are the same on the RGB and thermal images. The following *sec*tion reports mAP values and compares classification performance for the two models with the RGB and thermal datasets.

In this section, using the RGB dataset, it was shown that accuracy of prediction can significantly decrease when there are heavy debris and overlapping of peppers. In addition, the variant ambient lighting can hamper the performance of detection algorithms based on RGB data.

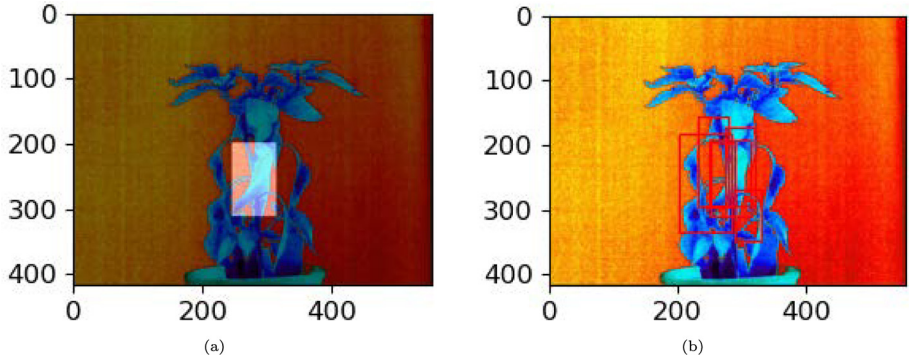


Fig. 22. Comparison of thermal Mask R-CNN: (a) Actual plot, (b) Predicted plot.

Table 4
YOLOv3 mAP values.

RGB		Thermal	
Train	Test	Train	Test
1.0	1.0	0.99	0.97

Performance of the Mask R-CNN during this portion of our investigation leads us to doubt the possibility of using the model for real-time applications with thermal imagery. The YOLOv3 model, however showed promising results. Table 4 displays RGB and thermal mAP values for the YOLOv3 architecture. There is a slight decline in performance from RGB to thermal; however compared to the decline that occurred with the Mask R-CNN architecture, this decline is acceptable.

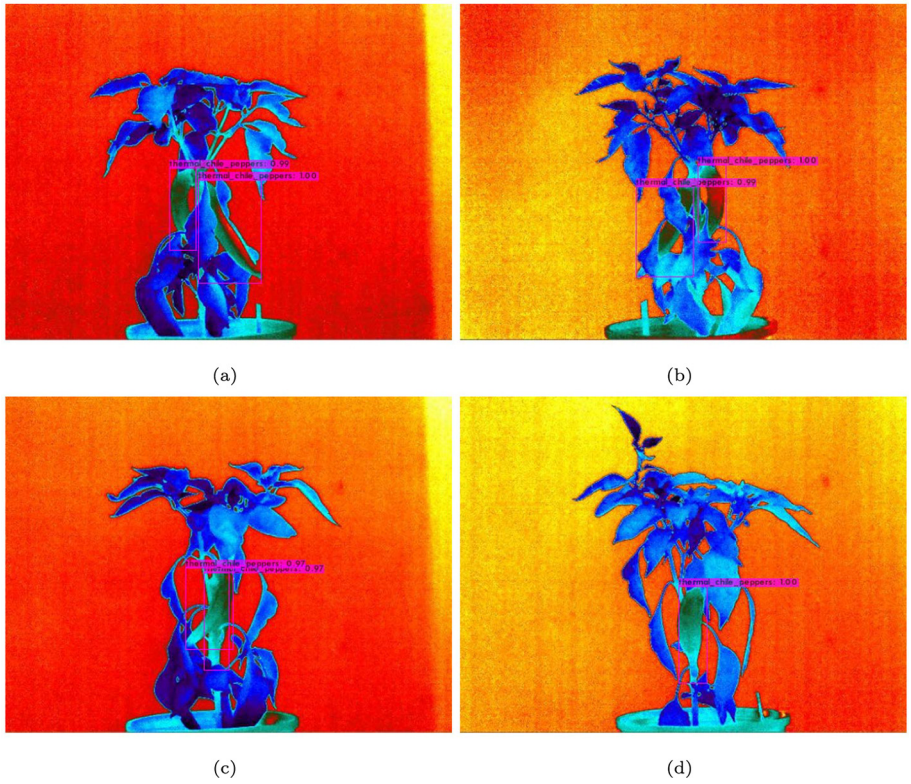


Fig. 23. Thermal chili predictions on validation dataset with YOLOv3: (a) Prediction test image 2, (b) Prediction test image 3, (c) Prediction test image 4, (d) Prediction test image 5.

In the following section, non-destructive thermal imaging will be used to resolve these issues as well as highlight results of the Mask R-CNN and YOLOv3 models used with the thermal dataset.

4.4. Thermal imaging

The same process conducted with the RGB dataset was used with the thermal dataset. 80% of the images were used for training with 20% held for testing. Predictions from the Mask R-CNN model did not benefit from thermal imaging and Prediction performance was decreased drastically. Table 3 displays the comparison of mAP values from the RGB and thermal imagery datasets showing precision reduced over half. We see testing mAP values drop by almost 0.6 for the Mask R-CNN, which is concerning since these models only predict on one class.

Fig. 21 shows how the masks were applied during training to a variety of different images. Fig. 21a demonstrates an image in heavy debris due to the pepper location directly behind the plant stem. Figs. 21b through 21d demonstrate peppers in light debris. We see how debris negatively affects prediction for the Mas R-CNN in Fig. 22. A detailed inspection of the figure reflects the tabular results by comparing actual and predicted plots of a test image. In Fig. 22a, the image shows one bounding box applied to a single pepper however, the Mask R-CNN model predicted 7 bounding boxes on the test image which decreases the mAP value significantly.

Fig. 23 demonstrates how thermal imaging can be useful for predictions made with debris. When we compare test images 2, 3, and 5 in Figs. 18 and 23 we can see that prediction scores do not significantly change. However, test image 4 shows a dramatic improvement in Fig. 23. In Fig. 18 (c) where there is overlapping of peppers, prediction scores for the peppers are reported as 1.00 and 0.67. Compared to the same pepper configuration, Fig. 23 (c) shows that both peppers are predicted at scores of 0.97.

Prediction performance of YOLOv3 with the thermal image dataset showcases the possibility of utilizing object detection for robotic harvesting with debris, paper overlapping, and low light present. Further testing should be conducted to determine the possibility of using object detection for robotic harvesting during evening and night hours.

5. Conclusion

In this study, utilized RGB and thermal images of chili peppers in an environment of various amounts of debris, pepper overlapping, and ambient lighting, train this dataset, and compare object detection methods. Results of feature visualization with HOG and object detection with Mask R-CNN and YOLOv3 architectures for a novel chili pepper image dataset were reported. HOG helped display gradient edge directions, and the Mask R-CNN model provided suitable prediction performance for less than real-time detection on the RGB dataset. The YOLOv3

algorithm has superior capabilities to the Mask-RCNN with over ten times the computational speed on the chili dataset. The YOLOv3 model introduces a spike in model performance for real-time detection (~3 s predictions) and demonstrates the possibility of being utilized during real-time robotic harvesting. However, we reported a lack of confidence in detection when heavy amounts of debris and paper overlapping are present in RGB images. In this study, it was shown that non-destructive thermal imaging can resolve these issues. In particular, YOLOv3 demonstrated an ability to detect peppers in areas with high debris and pepper overlapping with the thermal imagery; however the Mask R-CNN model did not detect well on the thermal imagery. It was shown that mapping temperature differences between the pepper and plant/debris can provide significant features for object detection in real-time and can help improve accuracy predictions with heavy debris, variant ambient lighting, and overlapping of peppers. In addition, successful thermal imaging for real-time robotic harvesting could allow the harvesting period to become more efficient and open up harvesting opportunity in the evening hours or low light situations.

Further investigation should be pursued to improve object detection when dense debris and poor ambient light are present. One issue in our experimentation is that only one class was used during training. This issue could lead to overfitting and could be the reason that YOLOv3 performed so well on the datasets. Additional classes and variations in imagery type should be utilized during training and testing to avoid overfitting. We plan to expand the number and type of images in the dataset and determine if the YOLOv3 (or other models) can exceed performance expectations during real-time robotic harvesting in dense debris and poor lighting environments with peppers in natural environment.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdulla, W., 2017. Mask r-cnn for Object Detection and Instance Segmentation on Keras and Tensorflow. https://github.com/matterport/MaskRCNN.

Bochkovskiy, A., 2020. Yolov3 With Darknet Framework. https://github.com/AlexeyAB/darknet.

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: optimal speed and accuracy of object detection. arXiv 2004.10934 (preprint arXiv:2004.10934).

Bosland, P.W., Bailey, A.L., Cotter, D.J., 1991. Growing Chiles in New Mexico. Guide H-New Mexico State University, Cooperative Extension Service (USA).

Common Objects in Context (COCO), 2015. Detection Evaluate. https://cocodataset.org/detection-eval. Online; accessed 19 January 2021.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). 886–893. IEEE. https://doi.org/10.1109/CVPR.2005.177.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. CVPR09, pp. 248–255.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2007. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

FAO, 2019. Food and Agriculture Organization of the United Nations Statistics Division. (ACCESSED 23 FEBRUARY 2021). http://www.fao.org/faostat/en/data/QC/visualize/.

Felzenszwalb, P., McAllester, D., Ramanan, D., 2008. A discriminatively trained, multiscale, deformable part model. 2008 IEEE Conference on Computer Vision and Pattern Recognition. 1–8. IEEE. https://doi.org/10.1109/CVPR.2008.4587597.

Gao, Z., Shao, Y., Xuan, G., Wang, Y., Liu, Y., Han, X., 2020. Real-time hyperspectral imaging for the in-field estimation of strawberry ripeness with deep learning. Artif. Intell. Agric. https://doi.org/10.1016/j.aiia.2020.04.003.

Girshick, R., 2015. Fast r-cnn. Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448. https://doi.org/10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al., 2018. Recent advances in convolutional neural networks. Pattern Recogn. 77, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

Hameed, K., Chai, D., Rassau, A., 2018. A comprehensive review of fruit and vegetable classification techniques. Image Vis. Comput. 80, 24–44. https://doi.org/10.1016/j.imavis.2018.09.016.

Hameed, K., Chai, D., Rassau, A., 2020a. A progressive weighted average weight optimisation ensemble technique for fruit and vegetable classification. 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, pp. 303–308.

Hameed, K., Chai, D., Rassau, A., 2020b. A sample weight and adaboost cnn-based coarse to fine classification of fruit and vegetables at a supermarket self-checkout. Appl. Sci. 10, 8667. https://doi.org/10.3390/app10238667.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37, 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969. https://doi.org/10.1109/ICCV.2017.322.

Jiang, J., Cen, H., Zhang, C., Lyu, X., Weng, H., Xu, H., He, Y., 2018. Nondestructive quality assessment of chili peppers using near-infrared hyperspectral imaging combined with multivariate analysis. Postharvest Biol. Technol. 146, 147–154. https://doi.org/10.1016/j.postharvbio.2018.09.003.

Kang, H., Zhou, H., Wang, X., Chen, C., 2020. Real-time fruit recognition and grasping estimation for robotic apple harvesting. Sensors 20, 5670. https://doi.org/10.3390/s20195670.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc, pp. 1097–1105. https://doi.org/10.1145/3065386.

Lenc, K., Vedaldi, A., 2015. R-cnn minus r. arXiv https://doi.org/10.5244/C.29.5 preprint arXiv:1506.06981.

Li, S., Luo, H., Hu, M., Zhang, M., Feng, J., Liu, Y., Dong, Q., Liu, B., 2019. Optical nondestructive techniques for small berry fruits: a review. Artif. Intel. Agric. 2, 85–98. https://doi.org/10.1016/j.aiia.2019.07.002.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125.

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R.J., Fredes, C., Valenzuela, A., 2020. A review of convolutional neural network applied to fruit image processing. Appl. Sci. 10, 3443. https://doi.org/10.3390/app10103443.

Patel, H.N., Jain, R., Joshi, M.V., et al., 2011. Fruit detection using improved multiple features based algorithm. Int. J. Comput. Appl. 13, 1–5. https://doi.org/10.5120/1756-2395.

Poirson, P., Ammirato, P., Fu, C.Y., Liu, W., Kosecka, J., Berg, A.C., 2016. Fast single shot detection and pose estimation. 2016 Fourth International Conference on 3D Vision (3DV). IEEE, pp. 676–684. https://doi.org/10.1109/3DV.2016.78.

Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271. https://doi.org/10.1109/CVPR.2017.690.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 779–788. arXiv preprint arXiv:1804.02767.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: towards real-time object detection with region proposal networks. arXiv https://doi.org/10.1109/TPAMI.2016.2577031 preprint arXiv:1506.01497.

Sadeghi, M.A., Forsyth, D., 2014. 30hz object detection with dpm v5. European Conference on Computer Vision. 65–79. Springer. https://doi.org/10.1007/978-3-319-10590-1-5.

tzutalin, 2017. Graphical Image Annotation Tool. https://github.com/tzutalin/labelImg.

Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W., 2013. Selective search for object recognition. Int. J. Comput. Vis. 104, 154–171. https://doi.org/10.1007/s11263-013-0620-5.

Vadivambal, R., Jayas, D.S., 2011. Applications of thermal imaging in agriculture and food industry—a review. Food Bioprocess Technol. 4, 186–199. https://doi.org/10.1007/s11947-010-0333-5.

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE https://doi.org/10.1109/CVPR.2001.990517 Pp. I–I.

Viola, P., Jones, M.J., 2004. Robust real-time face detection. Int. J. Comput. Vis. 57, 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb.

Viola, P., Jones, M.J., Snow, D., 2005. Detecting pedestrians using patterns of motion and appearance. Int. J. Comput. Vis. 63, 153–161. https://doi.org/10.1109/ICCV.2003.1238422.

van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors, 2014. scikit-image: image processing in Python. PeerJ 2, e453. https://doi.org/10.7717/peerj.453.

Wan, S., Goudos, S., 2020. Faster r-cnn for multi-class fruit detection using a robotic vision system. Comput. Netw. 168, 107036. https://doi.org/10.1016/j.comnet.2019.107036.

Yan, J., Lei, Z., Wen, L., Li, S.Z., 2014. The fastest deformable part model for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2497–2504. https://doi.org/10.1109/CVPR.2014.320.

Yu, Y., Zhang, K., Liu, H., Yang, L., Zhang, D., 2020. Real-time visual localization of the picking points for a ridge-planting strawberry harvesting robot. IEEE Access 8, 116556–116568. https://doi.org/10.1109/ACCESS.2020.3003034.

Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. Object detection in 20 years: a survey. arXiv 1905.05055 preprint arXiv:1905.05055.