computers

MDPI

Article

A Seamless Deep Learning Approach for Apple Detection, Depth Estimation, and Tracking Using YOLO Models Enhanced by Multi-Head Attention Mechanism

Praveen Kumar Sekharamantry [1,2,*], Farid Melgani [1], Jonni Malacarne [1], Riccardo Ricci [1], Rodrigo de Almeida Silva [3] and Jose Marcato Junior [3]

[1] Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy; farid.melgani@unitn.it (F.M.); riccardo.ricci-1@unitn.it (R.R.)

[2] Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam 530045, India

[3] Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande 79070-900, Brazil; rodrigo.a.silva@ufms.br (R.d.A.S.); jose.marcato@ufms.br (J.M.J.)

[*] Correspondence: pk.sekharamantry@unitn.it

Abstract: Considering precision agriculture, recent technological developments have sparked the emergence of several new tools that can help to automate the agricultural process. For instance, accurately detecting and counting apples in orchards is essential for maximizing harvests and ensuring effective resource management. However, there are several intrinsic difficulties with traditional techniques for identifying and counting apples in orchards. To identify, recognize, and detect apples, apple target detection algorithms, such as YOLOv7, have shown a great deal of reflection and accuracy. But occlusions, electrical wiring, branches, and overlapping pose severe issues for precisely detecting apples. Thus, to overcome these issues and accurately recognize apples and find the depth of apples from drone-based videos in complicated backdrops, our proposed model combines a multi-head attention system with the YOLOv7 object identification framework. Furthermore, we provide the ByteTrack method for apple counting in real time, which guarantees effective monitoring of apples. To verify the efficacy of our suggested model, a thorough comparison assessment is performed with several current apple detection and counting techniques. The outcomes adequately proved the effectiveness of our strategy, which continuously surpassed competing methods to achieve exceptional accuracies of 0.92, 0.96, and 0.95 with respect to precision, recall, and F1 score, and a low MAPE of 0.027, respectively.

Keywords: apple detection; depth estimation; multi-head attention mechanism; ByteTrack

check for updates

Citation: Sekharamantry, P.K.; Melgani, F.; Malacarne, J.; Ricci, R.; de Almeida Silva, R.; Marcato Junior, J. A Seamless Deep Learning Approach for Apple Detection, Depth Estimation, and Tracking Using YOLO Models Enhanced by Multi-Head Attention Mechanism. *Computers* **2024**, *13*, 83. https://doi.org/10.3390/computers13030083

Academic Editor: Paolo Bellavista

Received: 5 February 2024
Revised: 11 March 2024
Accepted: 18 March 2024
Published: 21 March 2024

1. Introduction

Apples are a major agricultural export across the world, contributing significantly to agricultural economic growth. Recently, computer vision-based systems have been employed in a wide range of applications, including biomedical [1,2], remote sensing, agricultural and farming monitoring, multimedia, and so on. The study's purpose is to develop a deep learning-based technology for agricultural automation. However, experienced farmers continue to be the driving force behind agricultural production. Manual labor wastes time and raises production costs, and workers with insufficient expertise and experience are prone to errors [3]. The advent of smart agriculture has fueled the integration of intelligence in orchards, which has emerged as a critical aspect in obtaining exact product information [4]. A visual system with automatic recognition based on support vector machine was proposed to identify fruit in orchards for autonomous growth evaluation, robotic harvesting, and yield calculation [5,6]. The vision system controlled the end result of the robot collecting apples from trees by identifying and localizing the apples.

As a result, detecting and tracking of apples is a critical challenge for these applications. Conversely, effectively recognizing fruits in natural situations poses substantial hurdles. Fruit detection can be inaccurate due to factors such as changing lighting conditions, overlapping shading, and similarities between distant little fruits and the backdrop. Many overlapping occlusions, leaf occlusions, branch occlusions, and other issues have also been identified, resulting in fruit target identification challenges that make fruits difficult to detect, recognize, and identify with high accuracy. The data collection of apples from farm fields is rather complex, as shown in Figure 1a–d.

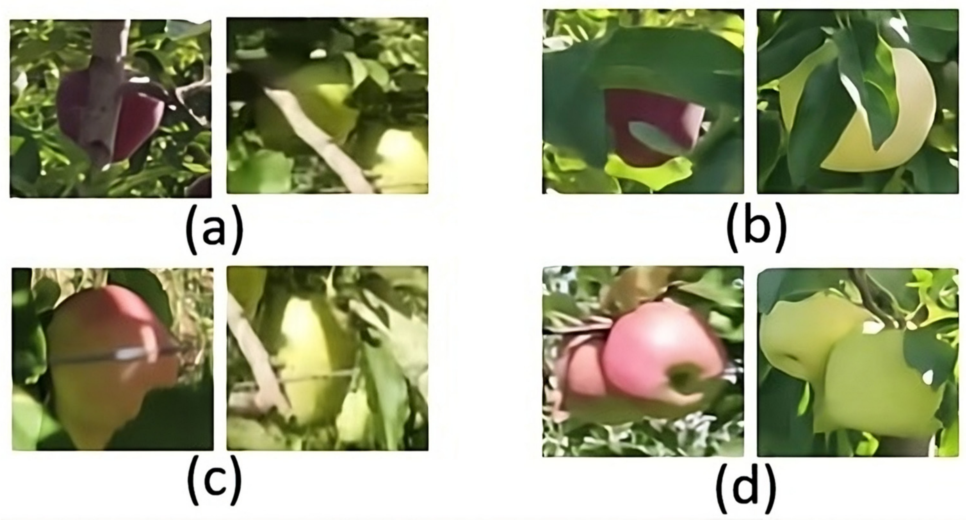


Figure 1. (**a**) Fruits concealed by branches; (**b**) Fruits obscured by leaves; (**c**) Fruits occluded by trellis wire; (**d**) Overlapping or bunched fruits.

In images, objects may overlap or be placed close together, with one object partially concealing the other. Obscured objects cannot be completely identified or annotated if occlusion is not handled correctly. Smaller or thinner objects, which have more of their surface area blocked, are more severely affected by occlusion. The concrete way of handling the problem would be to label the occlusion at the bounding box level to instruct the model as to which areas of an item are hidden. Thus, when producing a detection prediction, the model can then factor out the obscured characteristics. Also, image segmentation masks, bounding box overlays, and other techniques can be used to artificially occlude dataset objects. This demonstrates to the model how various items seem while partially obscured. The likelihood of missing or incorrectly packaging or labeling these difficult-to-see items is higher. Predicting the apple yield for a specific crop is a challenging task. For instance, an existing method for optimal thresholding for automatic recognition of apple fruits [7] claims that a low threshold of 0.2 has an extremely high recall. We face the possibility of obtaining too many false positives, which is the drawback. Similarly, according to Bin Yan et al. [8], a model will be extremely accurate with a threshold of 0.8, but the number of unrecognized apples will significantly increase. The most logical starting point would seem to be a threshold of 0.5 [9]. The detection of unidentified apples is much better with a confidence threshold of 0.5. The traditional methods are dedicated to the maturity of apples analyzed by the shape, size, and color of the apples [10,11] before detection and harvesting. Bulanon et al. [12] used threshold segmentation to improve the color difference of the red channel of the apple picture and extract the apple fruit target. The processing recognition rate reached 88.0%; however, it was only 18.0% in the backlight environment. Tian et al. [13] suggested a localization strategy based on depth information in pictures to determine the circular center, match the shape, and increase identification accuracy to 96.61%.

According to Lei Hu et al. [14], their proposed model offers an enhanced YOLOv5 algorithm for mature apple target detection in challenging situations. To improve accuracy and efficiency, it includes an adaptive scaling mechanism and a position focus loss function. To categorize and correlate the apple targets, the method uses the concept of feature information extraction and employs the position focal loss function. This helps to prevent feature information loss while also improving the algorithm's accuracy and efficiency. The new algorithm displays an 8.1% improvement in accuracy and 3.9 frames per second increase in pattern recognition speed through experimental examination of apple target feature data under varied conditions. The suggested method provides a solution for effectively detecting and locating ripe apples in complicated surroundings, which is useful for apple-picking robots and other applications. In spite of this, the study does not go into great depth about the adaptive scaling technique and position focus loss function employed in the enhanced YOLOv5 method. The report does not specify which complicated contexts were used to test and assess the new technique, nor does it compare the enhanced algorithm's performance to that of other current methods for detecting targets under challenging situations.

The work proposed by Jiuxin et al. [15] on apple-picking robots provides a quick technique for apple recognition and processing based on a modified version of the YOLOv5 algorithm. The enhanced model is easier to migrate and apply to hardware devices since it is smaller (57% smaller) and faster (27.6% faster) at processing data. The target association identification method increases efficiency by cutting the model selection process processing time by 89%. When compared to previous deep networks, the enhanced YOLOv5 model performs more quickly and accurately, making it a useful tool for apple recognition [16].

The lightweight MobileNetv2 network uses the inverted residual convolution module in place of the YOLOv5 backbone standard convolution module. The least-squares method is used to fix the model's inaccurate data output findings, making it better suited for distinguishing different apple forms. The approach of target association recognition is introduced when developing multi-target picking pathways according to the correlation among the confidence levels of the recognized targets. These methods are combined to enhance YOLOv5s, the model size is reduced, and the detection speed is increased, making it easy to migrate to and use in hardware devices. The suggested path planning method, which is based on the enhanced YOLOv5 model, lowers computation costs and successfully addresses the issues of processing massive volumes of information and repeating processing that arise throughout the apple picking activity. The target recognition information can be further utilized to provide suggestions for obstacle avoidance in the apple picking process.

An automated vision system was created employing stereo cameras synced to a customized LED strobe for on-tree measuring of apples in photos using excellent measurement precision [17]. Faster R-CNN and Mask R-CNN, two deep neural network models, were trained to find fruit candidates for size and extrapolate obscured fruit sections to enhance size estimation. The stereo cameras' spatial resolution and depth data were used to translate the segmented fruit shapes into metric specific surface areas and diameters. The camera system was used in monthly field tests from June to October to measure fruit size in the range of 22 to 82 mm and compare them to ground truth diameters. To determine the effect of fruit form on size estimation using images, a laboratory setting experiment was carried out. The 2D surface of an apple in an image, calculated in metric units that used the camera system, was used to describe fruit shape. In the experiment, altogether 100 apples (50 "Candy Crisp" and 50 "Rome") were imaged in various orientations to mimic field settings. In an analysis of the link between focal length, camera field-of-view, and size accuracy, it was discovered that increasing the distance from the tree reduced the pixel count and size accuracy. The imaging system delivered accurate measurements of fruit size and weight, as demonstrated by in-field comparisons of the readings with ground truth data. The study also included details on the dates and types of data acquired during the field experiment, including monthly image capture, ground truth size measurements, and fruit weight records. However, for fruit recognition and occlusion handling, the study used a specific pair of models of deep neural networks (Faster R-CNN and Mask R-CNN), and the effectiveness of other models or techniques was not examined, similar to the few traditional works on apple fields [18,19]. The use of stereo vision and machine learning in agricultural imagery may have drawbacks or difficulties, which include issues with illumination, image quality, and processing needs.

The YOLOv7-tiny-Apple model, which has been proposed as a lightweight small-target apple recognition and counting tool, can be used for autonomous orchard management, assisting in real-time apple detection and more efficient orchard management by identifying and counting apples [20]. The model provides theoretical support for developing apple identification and counting models by providing new insights on hardware installations and orchard yield estimation. It may be used in orchard management in real time to improve labor efficiency, product quality, and agricultural operational efficiency. The work makes use of the publicly available MinneApple dataset, which has been processed to create a collection of photos with diverse weather conditions, such as scenarios with fog and rain. The suggested detection algorithm is built on the updated YOLOv7-tiny model, which includes skip connections to shallower features, P2BiFPN for multi-scale feature fusion [21], and a lightweight ULSAM attention mechanism to minimize the loss of small target features. The suggested model, YOLOv7-tiny-Apple, demonstrated better detection accuracy with a mean average precision (mAP) of 80.4%, as well as a loss rate of 0.0316, which was 5.5% higher than the baseline model. The mean absolute error (MAE) was 2.737 and the root mean square error (RMSE) was 4.220 in terms of counts [22], which were 5.69% and 8.97% less than the original model, respectively. The amount of equipment needed was decreased by 15.81% due to the smaller size of the model. The suggested model showed improved generalization and resilience, making it appropriate for tiny target apple detection in a natural context with complicated backdrops and shifting weather conditions. The model needs to improve technological monitoring and management of smart orchards, lightweight optimization, greater detection accuracy, and mobile device deployment.

However, the efficacy of all of these systems is compromised due to backdrop complexity, motion blurriness, poor light, obstacle avoidance, and other factors. In this work, we propose a novel deep learning strategy based on the YOLOv7 model to address these concerns. In addition to this design, we have included a multi-head attention mechanism (MAM) technique to deal with size changes and predict the depth of apples in the orchard field. The following are the primary innovations and authors contributions of the proposed approach:

- To make our training dataset more effective, we included an attribute augmentation approach to offset the issue of contextual data loss and a feature improvement model that would enhance the representation of features and speed up inference.
- The YOLOv7 model is implemented on the augmented data for apple detection in live apple orchard fields.
- A multi-head attention mechanism is integrated with YOLOv7 to compute the depth of apples.
- The apples are tracked and counted using an enhanced ByteTrack technique.

The article is organized as follows: Section 2 deals with the proposed system in which data acquisition and data augmentation are applied on the dataset of apples to eliminate various factors of external sources that will affect the accuracy of the model. The detection of apples is dealt with by the improved YOLOv7 on the pre-processed data after data augmentation. A multi-head attention mechanism is applied to YOLOv7 to find the depth of apples along with the detection accuracy. Collaborating with these, the ByteTrack approach is finally used to track and count the number of apples. Section 3 presents the experimental results of the proposed methodology and a comparative analysis to demonstrate the robustness of the suggested method in comparison to standard object detection techniques. Section 4 clarifies the discussion of the methodologies and results obtained. Finally, Section 5 gives the conclusion and future potential of this seamless apple detecting and counting approach. By combining these elements, a complete system that can track and identify apples and comprehend the distance between apples and drones is obtained, opening up a valuable application.

2. Proposed Methodology

The recommended approach deals with a complete introspection of apple detection, finding the depth of apples and finally tracking and counting of the number of apples by the drone based on live apple orchard videos.

2.1. Data Acquisition and Pre-Processing

For the drone-cantered apple identification structure to be trained and evaluated, a refined custom dataset is required. The preparation process for data acquisition plays a vital role in the overall accuracy of the model. Reflex, stereo cameras, and a drone were used to gather the data on two distinct fields. The data collection was carried out in Val di Non in Trento, Italy, where the apple fields are situated, as shown in Figure 2a,b. The photos and videos were taken between the plants at a distance of 30 and 60 cm. The data were collected by the drone on a day in September with a variety of weather conditions, as shown in Figure 2c,d. There were no additional lights or artificial lighting used during the flight. The drone settings were processed via the rtmp protocol, which connects the camera to the drone's backend storage. It has an interference-free maximum transmission range of 80 m and height of 50 m.

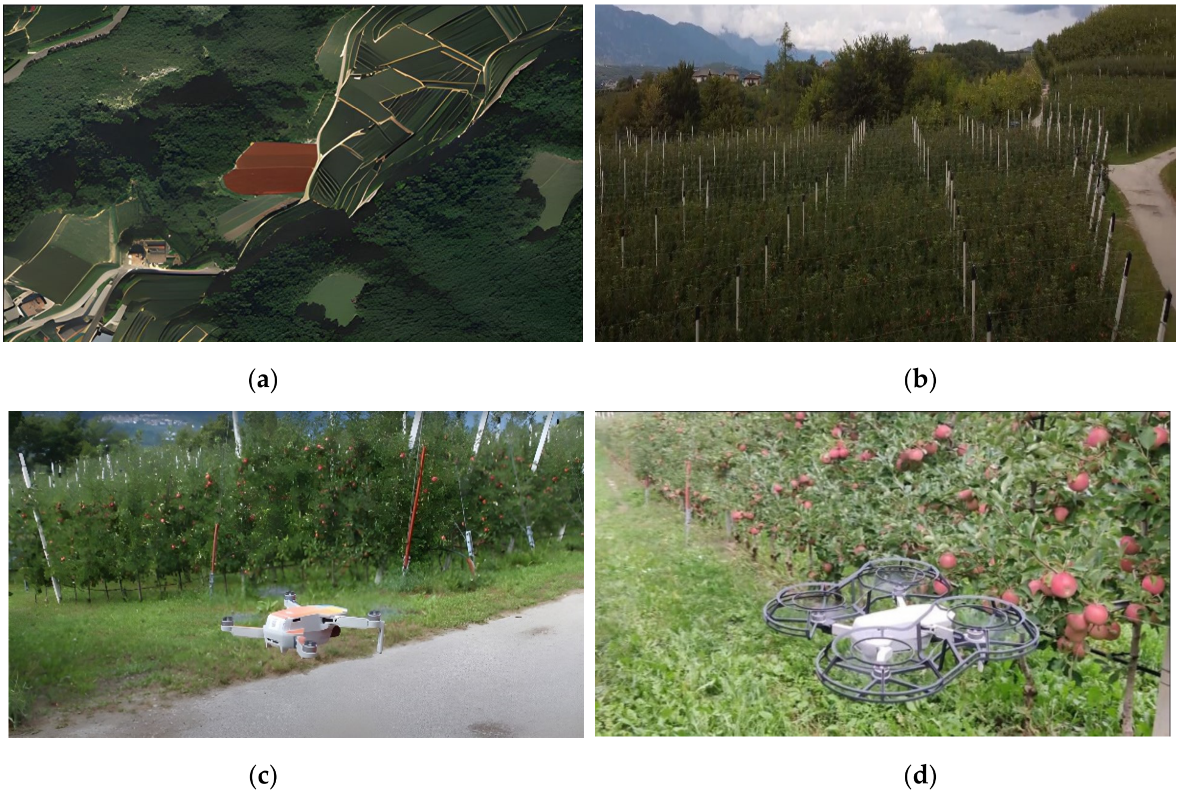


Figure 2. (**a**) Geomap location of the apple orchard; (**b**) Drone camera view of the apple field; (**c**) Drone flying in the apple field; (**d**) Drone camera recording the apples.

A DJI Mavic mini 3 drone and a Stereo Labs ZED 2iw Polarizing Filter were used. It has a 249 g ultra-light option and 5-kilometre HD video transmission, and it can record high-resolution drone videos. The drone has GPS-precise hover and a vision sensor. With streamlined recording and editing, the three-axis Gimbal 2688 × 1520 resolution camera provides a detailed image. The camera's field of vision is 44° in the vertical and 81° in the depth. To confirm the robustness of the model, additional material included different lighting situations, angles, and orchard layouts.

(a) Annotations: A fraction of the image are captioned by labeling the apples for each frame of the 10 GB entire footage, which is divided into many frames. Each apple has a bounding box drawn around it, and the depth labels indicate how close it is to the drone. The dataset is particularly confined to apples that are ready to harvest, and the immature apples are curtailed during the first labeling effort. As a result, our trained method will be able to distinguish only mature apples under varied environmental conditions. The ground truth data from the annotations are used to train and test the model's accuracy.

(b) Dataset classification: A training dataset, a testing set, and a validation dataset are created from the full dataset. A considerable portion of the images are from the training set, while only 30% and 10% of the images are from the testing and validation sets, respectively. The testing set evaluates the trained model's performance.

2.2. Data Agumentations

In computer vision applications, data augmentation [23] is a critical part since many elements must be taken into account when the data collected are affected by external sources. We noticed that the distance between the camera and the trees fluctuated during the process since several apple images were relatively small while others were pretty large. There was a significant asymmetry in the original data. Also, applications that operate in real time suffered from this sort of input data uncertainty. Hence, training with this type of unbalanced data may result in over-fitting and reduce detection accuracy. In a similar way, the apple image data collection mechanism faces same issues during image capture. As a result, data augmentation becomes an essential duty in these sorts of tasks where object sizes differ often. Hence, the following augmentation approaches were taken into consideration:

Image radiance: In order to match the actual low light and bad illumination circumstances, the brightness is alternately increased and decreased. With the function "hsv2rgb", the image is first converted to HSV and then to RGB.

Flipping of Image: To help the image classifier recognize apples in various situations, the vertical as well as horizontal pixels are mirrored.

Rotating the image: When the capturing angle is constrained, the drone angle is not fixed. So the model needs to be trained to be capable of capturing and identifying the apple from a variety of perspectives.

Image blur: The drone moves at various speeds, and the video frequently records ambiguous information. The model can be trained using blurry images to help with accurate detection standards.

Noisy image: Images are subjected to a standard 0.02 of Gaussian variance [24,25]. High heat and electronic circuit noise may be produced by the drone. By utilizing Gaussian noise, this process would assist in creating a model of human motion with human qualities.

The cautiously improved dataset serves as the foundation for developing and testing the suggested drone-based apple recognition system. It consists of annotated drone-taken images of apples with corresponding depth labels.

2.3. YOLOv7 Architecture

The YOLOv7 model is a current-time object identification system for detecting apples in pictures or videos. YOLOv7 is an improved variation of the popular You Only Look Once (YOLO) approach, which estimates box boundaries and probabilities of classes for each object in a picture [26–28]. The YOLOv7 algorithm provides the best accuracy of any real-time object identification model while maintaining 30 frames per second or more. It uses far less hardware than conventional neural networks and therefore can be trained considerably more quickly on tiny datasets with no pre-learned weights. Researchers have presented many approaches for detecting apples utilizing the YOLOv7 model. One study, for example, developed a better method built on the YOLOv7 model to solve the poor performance of apple fruit detection due to the complex backdrop and occluded apple fruit. Other research employed an updated YOLOv7 framework and multi-object tracking algorithms to recognize and count apples in apple orchards [29,30]. The approach dealt with transformers to determine apple ripeness from digitized photos of several apple varieties.

In general, the YOLOv7 algorithm provides a robust tool for identifying apples in various contexts, and researchers are always looking for new ways to increase its precision as well as efficacy. The model design as well as the training method were optimized using YOLOv7. In model architecture, YOLOv7 provides an expanded, adequate layers aggregation network and scaling model skills. During the training phase, YOLOv7 replaces the original module with model re-parameterized skills and employs a dynamic label assignment technique to apply labels to distinct output layers. The standard YOLOv7 model's architecture for detection of apples involves basic components such as the inputs, backbone, neck, detection heads, and prediction output, as shown in Figure 3.

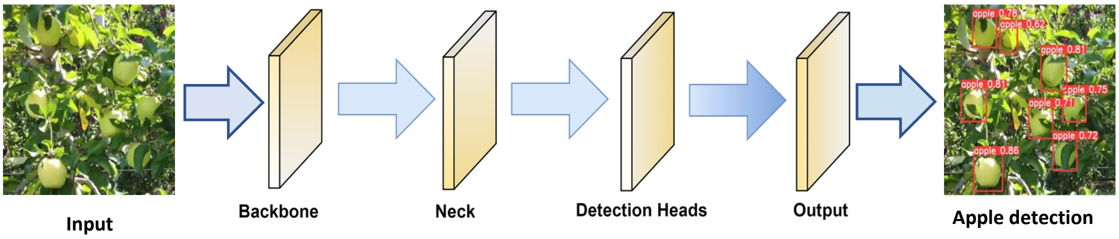


Figure 3. Basic description of YOLOv7 architecture.

The initial process in the procedure is to analyze the input image, which comprises different variants of apples images that need to be identified. The selection of a proper backbone network is essential for apple detection. The backbone network passes the input image through a number of convolutional layers. Specific filters are applied with the help of the convolution layer to the source images to capture varied features at diverse spatial resolutions. For object boundary detection, the network will first learn to recognize basic features like edges and colors in the first layers. The hierarchical feature learning is processed by the backbone network deeper layers by picking up on more complicated and abstract properties. At these deeper layers, features like texture, patterns, and object portions are learned, enabling the network to comprehend the fine details of apples. Higher-level semantic characteristics are extracted as the image is being processed through the backbone.

To identify apples from other items, these traits encode characteristics about object shape. One of the best features of the backbone is that it is made to be resistant to variations in scale and rotation. As a result, the network is able to recognize apples in the input image regardless of their dimension or orientation. The backbone network produces feature maps that spatially map the learned features on the image. Subsequent layers use these feature maps, which are packed with data about the input apple images, to detect objects. The network neck receives the feature maps that were retrieved from the backbone. The neck further enhances these features, frequently forming a feature pyramid that aids in the detection of objects of various sizes. The detection head processes the feature maps at the end and predicts the bounding box dimensions and class probabilities for the discovered apples.

The neck's role is to build a pyramid structure in which lower-resolution maps are obtained from higher-resolution ones (having finer information). As apples can exist in images in a variety of sizes, their pyramidal structure is crucial. The network can efficiently detect both large and small apples since it has features at many scales. Further feature fusion aids in capturing contextual data for apple detection. For instance, it enables the network to recognize how an apple interacts with its environment, facilitating precise detection. The neck also deals with the contextual information. Apples can be distinguished from other items and backgrounds using contextual information [31]. As an illustration, the existence of leaves, branches, or specific colors around an apple can serve as helpful detection cues. The neck network improves the semantic understanding by recognizing the whole shape of apple roundness and other unique structural properties. The enhanced and refined attributes from the neck are then handed to the head. These characteristics are used by the detection head to forecast apple-specific bounding box dimensions and class probabilities. The information from the neck is essential for the detection head to accurately estimate the location of the apples in the input image. Accuracy, real-time performance, good recognition efficiency, scalability, and effective hardware utilization are just a few benefits of using YOLOv7 to detect apples. Because of these benefits, YOLOv7 is a good choice for apple detection in a variety of applications, including monitoring systems and automated vehicles.

2.4. Improved YOLOv7 Architecture with Multi-Head Attention Mechanism

The improved YOLOv7 architecture in this section deals with the integration of the multi-head attention mechanism aimed at apple detection [32]. The modified framework accurately predicts the depth of apples and their confined features. The architecture diagram shown in Figure 4 depicts the addition of the multi-head attention mechanism within the framework of YOLOv7.

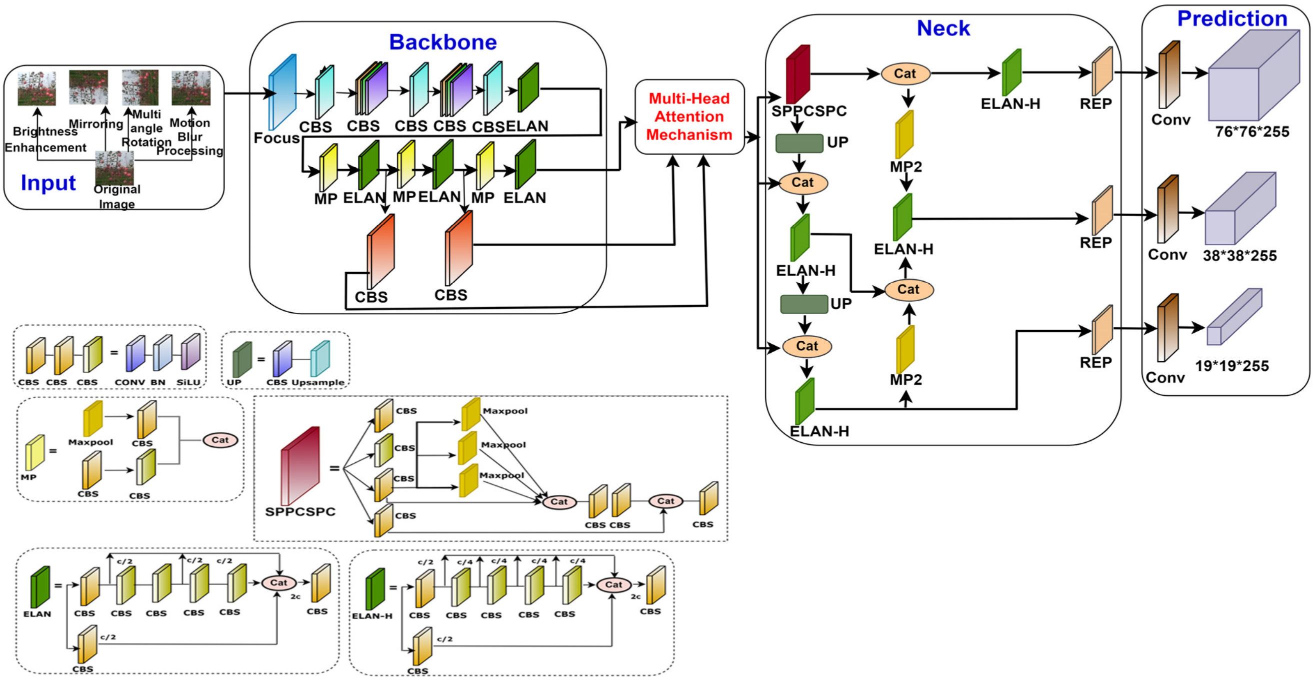


Figure 4. Architecture of YOLOv7 with multi-head attention mechanism.

In the architecture, the CBS layer performs the convolution, normalization of batches, and SiLU activation operations, which is the fundamental convolutional unit in the backbone. The feature map output of the ELAN (efficient layer aggregation network) layer is divided into three sections and is composed of several CBS structures. Here, the channel divides the feature map into two equal groups. The initial group then applies five convolution processes to produce the first component, the second group applies one convolution process to obtain the second one, and the third part is made up of the outputs of the first group's first convolution and third convolution. The feature map is divided into two groups by the MP (Max Pool) layer. Maximum pooling is used by the first group to extract more crucial information, and convolution is used by the second group to extract feature information. The outcome is finally obtained by joining two groups [33]. The CSPNet (convolutional spatial pyramid) including an SPP (spatial pyramid pooling) block makes up the SPPCSPC (spatial pyramid pooling and convolutional spatial pyramid pooling) layer. The CSPNet is a particular kind of network that incorporates data from several scales and resolutions to increase the detection precision. Spatial pyramid pooling, a technique used by the

SPP block to gather more contextual data, involves combining characteristics at several measures. The REP layer is a revolutionary idea that uses structural re-parameterization to modify the framework in inference to enhance the model performance. The REP layer can obtain the output of the feature map in three sections during training. Convolution and batch normalization are implemented in the first and second phases and only batch normalization is implemented in the third phase Structural re-parameterization uses less computational power, and model performance is enhanced as REP inference only keeps the second portion of the structure.

Multi-Head Attention Mechanism

In the real-world scenario, tasks capturing long range dependencies and their respective contextual data often become crucial. So, integrating the YOLOv7 model, as shown in Figure 5, with the multi-head attention mechanism offers a great advantage to deal with such problems. Convolutional neural networks (CNNs) that have undergone prior training can extract feature maps from input images. As an attention mechanism is incapable of identifying the spatial relationships between pixels, positional encoding must be added to the feature maps to provide the details about the positions of the image core components. Techniques like 1 × 1 convolutions can be used to lower the dimensionality of the features.

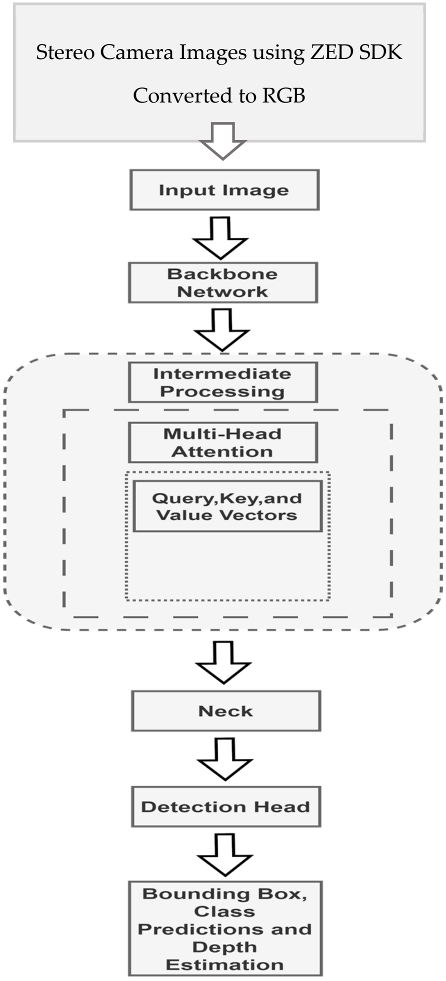

Stereo Camera Images using ZED SDK
Converted to RGB
Input Image
Backbone Network
Intermediate Processing
Multi-Head Attention
Query,Key,and Value Vectors
Neck
Detection Head
Bounding Box, Class Predictions and Depth Estimation

Figure 5. YOLOv7 architecture for apple detection and depth estimation.

The stereo cameras used in this work can capture high-resolution 3D video footage of apple orchards and determine depth by comparing the pixel displacement among the left and right pictures. Their two eyes are spaced 6 to 12 cm apart [34,35]. For every pixel (X, Y) in the picture, the ZED's depth maps record a distance value (Z). Measuring from the rear of the camera's left eye to the scene object, the distance is given in metric units (meters, for example). The ZED's default depth-detecting mode is called standard. The standard mode operates more quickly while maintaining distance metrics and shapes. We used the ValidMeasure function to determine valid depth data. Additionally, the ultra-depth mode provides computer vision-based techniques with the widest depth range and the best retained Z-accuracy across the sensing range. A feature called depth stabilization merges and filters the depth maps over many frames in a temporal manner. This makes it possible to reduce jitter and enhance the accuracy of depth on stationary objects like apples. By utilizing the ZED SDK's positional tracking feature, depth stabilization is still effective despite the fact that the camera is moving. To prevent merging the depth of dynamic regions, it may also identify moving objects. The depth resolution of a stereo camera varies across its range, and the formula Dr = Zˆ2*α describes how stereo vision employs triangulation to infer depth from a disparity image, where Dr is the depth resolution, Z is the distance, and α is a constant. The ZED SDK and accompanying tools are the only programs that can read the proprietary SVO file format. Together with metadata like timestamps and IMU (inertial measurement unit) data, it includes the camera's raw photos. Multiple file types can be created from SVO files for applications elsewhere. SVO may be exported into several formats using the sample ZED_SVO_Export SDK.

The source feature maps are linearly altered into several sets of queries, keys, and values. The input data maps are captured in a variety of ways by separately calculating the scaled dot-product attention scores. A residual connection is added from the input to the output of the multi-head attention. Also, a layer normalization is applied to make the training process more stable and efficient. After multi-head attention, the feature maps are trained over position-wise feed-forward neural networks. ReLU (rectified linear unit) activations and fully connected layers make up these networks. The network can more successfully capture the apples of different sizes because of multi-scale feature fusion. The detecting head predicts object class probability, bounding box coordinates, and other data required for object detection. Non-maximum suppression (NMS) is applied to exclude repetitive detections and choose the most certain predictions [36,37]. The model gains the ability to identify apples, forecast the bounding boxes, allocate class probabilities, and calculate their depths by minimizing the gap between estimated parameters as well as the ground truth labels.

The multi-head attention model is made up of the three components, query, key, and value, as shown in Figure 6. These components allow the model to concentrate on different input locations and collect relevant data. The concern is a representation of the area of interest that requires attention. A single feature or a collection of features describing an area in the feature maps connected to apples may be the query for apple detection. A query vector is created and then applied to every position in the input sequence. When compared to the key vectors, all query vectors are utilized to calculate the attention scores. Keys can represent either specific features that assist the model to recognize context, such as features from surrounding objects or regions, or features from the whole input image. To calculate the resemblance between queries and keys, key vectors are employed. If there is a lot of similarity, it means that the related portions of the data input need to be addressed.

The value, in accordance with the attention mechanism, refers to the properties that have been weighed and aggregated. The context of apple detection may benefit from the features that provide specific information about the recognized apples or their surroundings. Each attention head in the multi-head attention mechanism is in charge of mastering a different selective attention pattern or capturing a different aspect of the input material [38–40]. According to the calculated attention scores, value vectors are combined. Higher attention ratings indicate that the model places greater trust in the associated values when making predictions. Each attention head performs the computations for the query, key, and value separately. The weighted sum of the associated value vectors is computed using the attention results achieved from the SoftMax normalization. This weighted total, which reflects the focused information, is the result of each query attention mechanism.

The model can determine which areas of the images are important for identifying apples by employing attention techniques. For instance, the attention mechanism could assist the model in focusing on the visible parts of an apple if the apple is partially obscured by another object, increasing detection accuracy. In contrast, queries, keys, and values work together to create a multi-head attention mechanism that allows the model to dynamically focus on different elements of the input data. This feature is especially useful for detecting apples in complex scenarios.

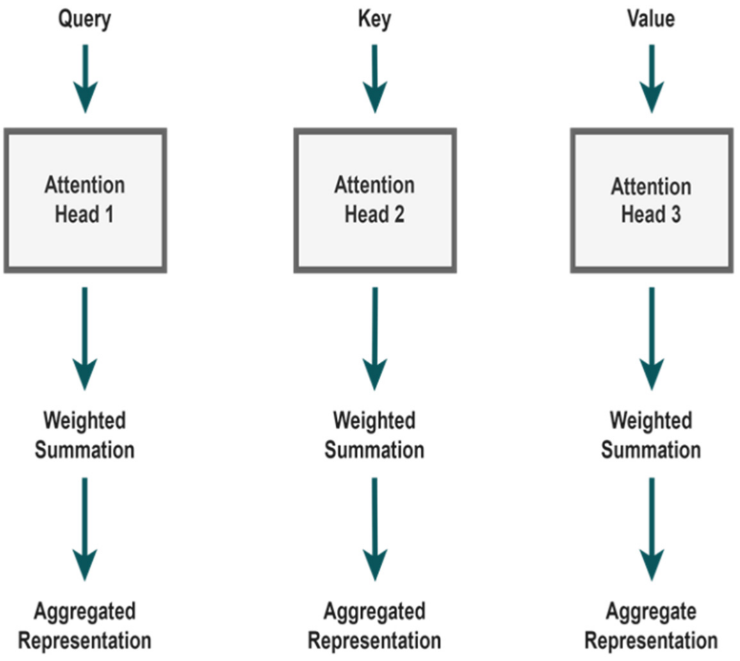


Figure 6. Components of multi-head model.

2.5. Box Prediction and Loss Function

The proposed YOLOv7 architecture neck module, which is defined above, is responsible of bounding box prediction. The ground truth of the bounding box is shown as W = (x_1, y_1, x_2, y_2). With these coordinates [41], Equation (1) is applied to determine W boundaries, as follows:

$$\begin{aligned} t_{\,x_1} &= log\frac{(s_l(x+0.5)-x_1)}{r_l}\ ,t_{\,y_1} = log\frac{(s_l(y+0.5)-y_1)}{r_l}\ , \\ t_{\,x_2} &= log\frac{(x_2\ \ -s_l(x+0.5))}{r_l}\ \ ,t_{\,y_2} = log\frac{(y_2\ \ -s_l(y+0.5))}{r_l} \end{aligned} \quad (1)$$

The ground truth boxes and projection coordinates are taken into account to calculate the normalized offsets between the coordinates, where s_l is the scaling factor, r_1 is the basic scale, and the coordinates of the image (x, y) are subsequently mapped to the original picture by applying down sampling. Using the log-space function at this point, we incorporate regularization. Later, the loss function is trained using the smooth L_1 loss function, and the bounding box prediction is performed using L_{reg}. Through iterative optimization, the loss function increases the accuracy of the target detection [42]. Classification and regression are the two primary components of the target loss detector loss function. The classification loss L_{cls} is among confidence, whereas the regression loss is in between the normalized border and regression target. The loss function is articulated as follows in Equation (2):

$$\begin{aligned} &L(\{p_{sl}\},\{t_l\}) = L_{cls} +\ L_{reg}\,\frac{1}{N_{cls}}\ \ \textstyle\sum_1 L_{cls}\ (p_{sl},p_l) + \lambda\ \frac{1}{N_{reg}}\textstyle\sum_1 p_l\ L_{reg}(t_i\ ,t_l) \\ &where = \begin{cases} \ \ p_i\ tf\ p_i = 1 \\ 1 - p_i\ otherwise \end{cases},\ \alpha_s = \begin{cases} \alpha\ \ if\ \ p_i = 1 \\ 1 - otherwise \end{cases} \text{ and C} = \begin{cases} 1\left|t_{ij}\ - t_{ij}\right| < 1 \\ \ \ 0\ otherwise \end{cases} \end{aligned} \quad (2)$$

Here, α is utilized to correct the positive and negative sample imbalance that results from the target image having fewer samples than the overall image, i.e., there are fewer samples of apple images than there are of the complete images. As a result, the model obtains accurate bounding boxes, which improves accuracy. The proposed focal loss function aids in estimating the classification loss, and α function is utilized to balance the effects of the proposed positive and negative loss functions. Additionally, it prevents the samples from producing a dominant amount of classification loss. L_1 loss is used for estimating the regression loss in order to determine the bounding boxes, and β aids in choosing the L_1 or L_2 loss function depending on the range of the loss. Furthermore, N_{reg} and N_{cls} regularize these loss functions. In order to construct the final optimal model, the overall loss L is propagated backward in a gradient way. In the process of apple detection, YOLOv7 + MAM is trained to identify apples in pictures or videos. Bounding boxes and class labels surrounding the identified apples, together with an estimate of their depth, will be generated as output by the model.

2.6. *ByteTrack*

Multiple object tracking (MOT) is a vital computer vision task that involves recognizing how various apples move over time in a video clip acquired from a drone. The objective is to identify, locate, and track every apple in the video, even when they are partially or entirely occluded by other elements of the scene.

Typically, there are two processes involved in multiple object tracking, object detection and object association, as shown in Figure 7. With object detectors like Faster-RCNN or YOLO, object detection is the process of recognizing all possible objects of interest within the present frame. Object association is the method of connecting tracklets or objects found in the current frame with their corresponding tracklets from earlier frames. Despite significant advancements, MOT is still a difficult task. There are some crucial problems that have prevented high-quality performance and contributed as the foundation for current methods.

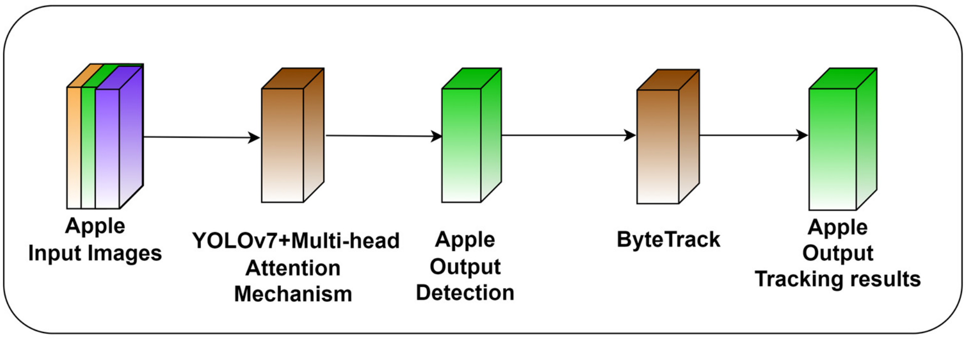


Figure 7. Proposed structure of multi-head detection and tracking of apples.

The visual input itself may cause complications. For instance, a single object motion and look can change significantly over the video sequence. Items in a scene can move in a variety of directions and at varying speeds. They can also alter in shape or size, as well as be completely or partially obscured by other objects. Several issues, such as object ID swapping or assigning numerous tracklets to the same item, lead to MOT tracking errors. Also, every apple object in the current frame must be consistently connected to its equivalent object in the previous frame, and the tracking system should be able to handle these deviations. A practical problem is the video inference speed while performing live video inference on apple orchards.

In our case, relying simply on a detection model makes counting unreliable because there is a significant chance the model may record numerous counts of identical apples if they occur in subsequent frames. Duplicate counts will lead to more false positive cases, which will reduce the model effectiveness and dependability for commercial application. The counting strategy should therefore be based on a method that is not only reliant on detections. So, a reliable method is to use an object tracking mechanism to follow each apple during the course of the video until the counter is incremented. According to the tracking-by-detection paradigm, a multi-object tracker (MOT) keeps track of numerous items of interest by detecting them in each time frame (t), connecting them to objects that were present in the previous frame ($t - 1$), and predicting their position in the next frame, (t + 1), thus tracking the items throughout time by repeating for each frame of the video sequence. The state of the object is predicted and updated using a Kalman filter in basic MOT methods like SORT [43], and the objects are associated using a Hungarian algorithm. The result of the MOT is bounding boxes with an ID produced specifically for each object to aid in object identification. However, these models may be prone to ID switching. As a result, the multi-object tracking accuracy (MOTA) is measured to assess the MOT's accuracy, as shown in Equation (3):

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t GT_t} \tag{3}$$

In this case, the terms FN, FP, IDS, and GT, respectively, refer to false negative, false positive, ID switch, and ground truth counts.

Building of ByteTrack

ByteTrack can resolve this issue by employing a motion model that controls a queue, called tracklets, to store the objects being tracked and conducts tracking and matching among bounding boxes having low confidence values. The main advancement of ByteTrack is the retention of non-background low confidence detection boxes, which are generally destroyed after the initial filtering of detections, and the use of these low-score boxes for a subsequent association phase [44,45]. Occluded detection boxes typically have confidence ratings that are below the threshold but still contain some information about the objects, giving them a better confidence score than background-only boxes. So, throughout the association phase, it is still important to maintain track of these low confidence boxes.

After the detection phase, the detected bounding boxes are filtered with preset upper and lower thresholds into high level of confidence boxes, low level of confidence boxes, and background boxes. After this procedure, background boxes are eliminated, but low- and high-confidence detection boxes are preserved for subsequent association stages. The detection accuracy boxes of present frames are matched with estimated boxes from previous frame tracklets (using Kalman filter) [46,47], which contain all active tracklets and lost tracklets from current frames, in a manner similar to normal association stages from other algorithms. The feature embeddings are matched using a simple IoU score or cosine similarity score (using feature extractors such as DeepSORT, QDTrack, etc.) using nearest neighbor distance and the Hungarian method or matching cascade [48,49]. Only if the similarity score exceeds a predetermined match threshold is the linear allocation among groups of bounding boxes confirmed. In the real implementation, mismatched high-score detection boxes are matched with tracklets that have updates from a single image before even being assigned to a new tracklet, as shown in Figure 8.

In the next step of association, the leftover unmatched predicted boxes of earlier frames are compared against low-score detection boxes. As it makes sense that obscured boxes should be less well linked to boxes from earlier frames, the matching method is the same as the first association step; however, the matching threshold is scaled lower. Unmatched detecting boxes are deleted, whereas unmatched prediction boxes are given the label lost tracklets. Prior to Kalman filter prediction, the lost tracklets are stored for a certain time of frames and added to the active tracklets. This enables the trackers to retrieve certain tracklets that were lost as a result of objects briefly going completely missing for a limited number of frames. The basic detector in the present work is YOLOv7. Users can choose from a variety of matching measures among IoU and ReID, depending on the characteristics of the datasets. The initial phase identification of high-score detections can be performed using either IoU or ReID. ReID performs best on videos with low frame rates or videos with noticeable frame-to-frame motion, whereas IoU is more trustworthy in extreme occlusion situations when ReID characteristics are unreliable. Consequently, second phase association should always employ IoU as the matching criterion since we can expect that low-score detection boxes will contain occluded apples with ReID features that might not be accurate depictions of the objects.

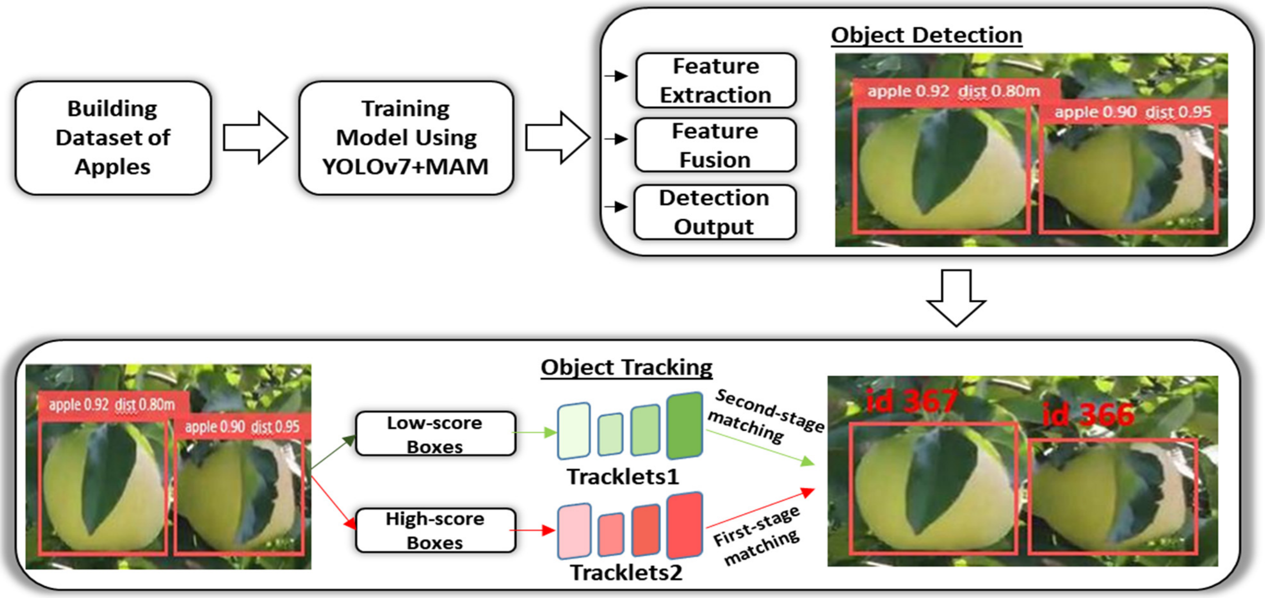


Figure 8. Proposed build of the model for detection and tracking of apples.

3. Results

The results of the suggested YOLOv7 + MAM architecture are shown in this section. Using real-time videos and images of apple orchards, this model was tested. The suggested method for apple tracking and detection was built with the Ubuntu 22.04 Linux operating system with the aid of the PyTorch deep learning framework. The operating system was installed with an Intel i7 processor, 24 GB of RAM, and an NVIDIA GeForce RTX 3090 linked to a 384-bit memory interface. The GPU operated at a rate of 1395 MHz. The Python programming language was used to develop the entire model. This model used YOLOv7, which was enhanced with a multi-head attention framework along with tracking using ByteTrack, and each model's performance was assessed with the aid of the CUDNN library and CUDA toolkit. The entire experiment was run with an IoU threshold set at 0.75.

The processing time examined in the current study is the duration the algorithm takes to analyze each frame of the drone-recorded input video stream. By contrast, memory usage is the amount of GPU-enabled system memory required for algorithm execution. The computational cost of our solution was calculated using a GPU-enabled computer system. We built the algorithm in Python and used the OpenCV package to process images. To determine the processing time, we measured the total execution time acquired by the algorithm on the video frames. Using the Python 'time' package, we recorded the start and end times of the processing pipeline and determined the average processing time per frame. Memory usage was analyzed using the 'memory_profiler' Python library, which allowed us to monitor the algorithm's memory consumption throughout execution. We measured peak memory consumption and averaged it across numerous iterations for a representative measure. Our computational cost evaluation revealed a 20 millisecond average processing time per frame and 2 millisecond standard deviation. Peak memory utilization during algorithm running was found to be around 300 MB.

3.1. Performance Assessment

The performance of the proposed approach was assessed using three indicators of accuracy: precision, recall, and F1 score. The following Equation (4) was applied to compute these parameters:

$$P_r = \frac{T_P}{F_P \ + T_P} \text{ , } Rec = \frac{T_P}{T_P \ + F_N} \text{ , } F1 - score = \frac{2P_r Rec}{P_r + Rec} \tag{4}$$

where T_P, F_P, and F_N represents the true positive, false positive, and false negative values.

3.2. Performance of Apple Detection

The results of the suggested approach to identify each apple in the picture are provided in this section. The performance that was attained was compared with that of the remaining models. To demonstrate the resilience of the suggested approach, we have considered several variables that impact system performance, like variations in illumination. We employed the PIL library in Python to account for changes in illumination, assigning a 0.5 factor for images with low brightness and 1.5 factor for pictures with high definition. The comparative study of the suggested method in terms of recall, precision, and F1 score for the original picture is shown in this section. The performance of the suggested approach was compared to that of several existing systems, including Faster RCNN, AlexNet With Faster RCNN, ResNet + FasterRCNN, YOLOv3, YOLOv5, YOLOv7, and finally with YOLOv7 + MAM. Table 1 displays the comparison outcomes for the detection performance.

Table 1. Performance comparison for the original images and under different lighting conditions.

	Original Images			**Illumination Variations**		
Detection Method	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Faster RCNN	0.84	0.78	0.80	0.68	0.72	0.71
AlexNet + Faster RCNN	0.88	0.83	0.86	0.69	0.75	0.71
ResNet + FasterRCNN	0.87	0.64	0.74	0.72	0.75	0.72
YOLOv5	0.82	0.86	0.84	0.71	0.77	0.73
YOLOv7	0.83	0.91	0.86	0.82	0.85	0.84
YOLOv5 + MAM	0.88	0.94	0.92	0.85	0.91	0.87
YOLOv7 + MAM	0.92	0.96	0.95	0.87	0.93	0.89

Figure 9a illustrates a comparison bar chart of original images with the different available methods and Figure 9b illustrates a comparison bar chart of illuminated images with the different available methods. The performance of the proposed methodology in detecting apples in live orchards is depicted in Figure 10. Figure 10a illustrates a straightforward frame of apples from an input video that was shot by a drone. Figure 10b presents the results of the detection of apples with YOLOv5 along with the multi-head attention mechanism. Figure 10c demonstrates the outcome of the improved detection of apples using YOLOv7 with the multi-head attention mechanism.

In comparison between the YOLOv5 + MAM model and YOLOv7 + MAM model, as shown in Figure 10, the number of apples identified by the proposed model was enhanced by comparing the output images. A few apples were undetectable and unidentifiable by the previous models. This problem was completely resolved by the proposed YOLOv7 + MAM model. Every apple had its depth displayed, which made it easier for us to see how far apart the apples were from one another and from the drone's spatial configurations. Taking into account the depth, the basic three-dimensional image of the apple from a different perspective would offer an estimate of apple yield. Increasing the localization accuracy is feasible to deal with occlusions. The results of estimating depth and detecting every potential apple are presented in the outputs. A few ground truth problems, such as sunlight and shade, can be fixed by altering the confidence and non-maximum suppression threshold. The model's accuracy was tested under various illumination conditions, like low, normal, and high illumination, and the proposed model accuracy was optimal under all conditions, as shown in Figure 11. Each bar represents a distinct environmental condition and the rise of the bar signifies the model's improved accuracy under varied conditions. The obtained performance was compared with the other models' performances. The agility of the proposed architecture was demonstrated by evaluating a number of factors that affect system performance, including noise, light change, and blurry images. Using a kernel of (3 × 3), the Gaussian blur method was used in the blurriness stage.

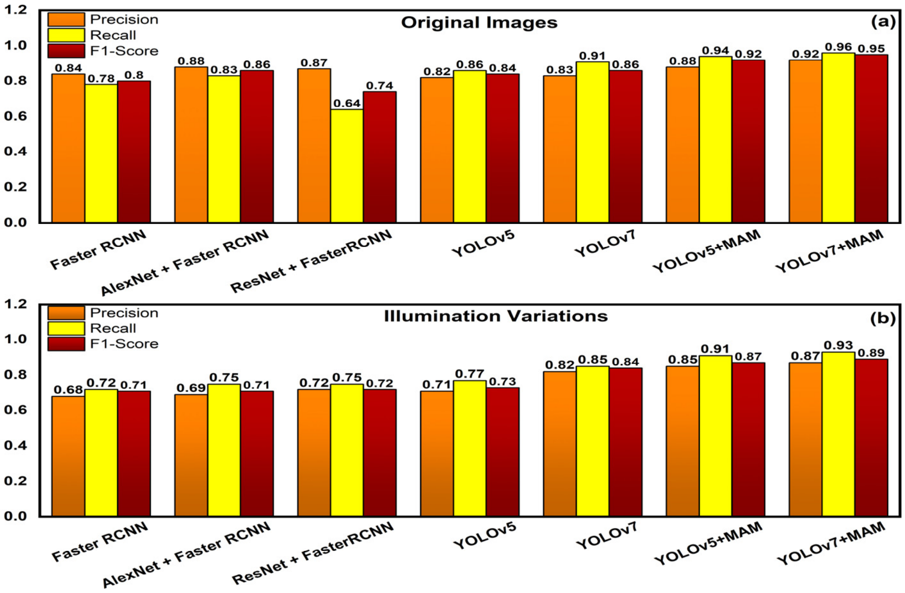


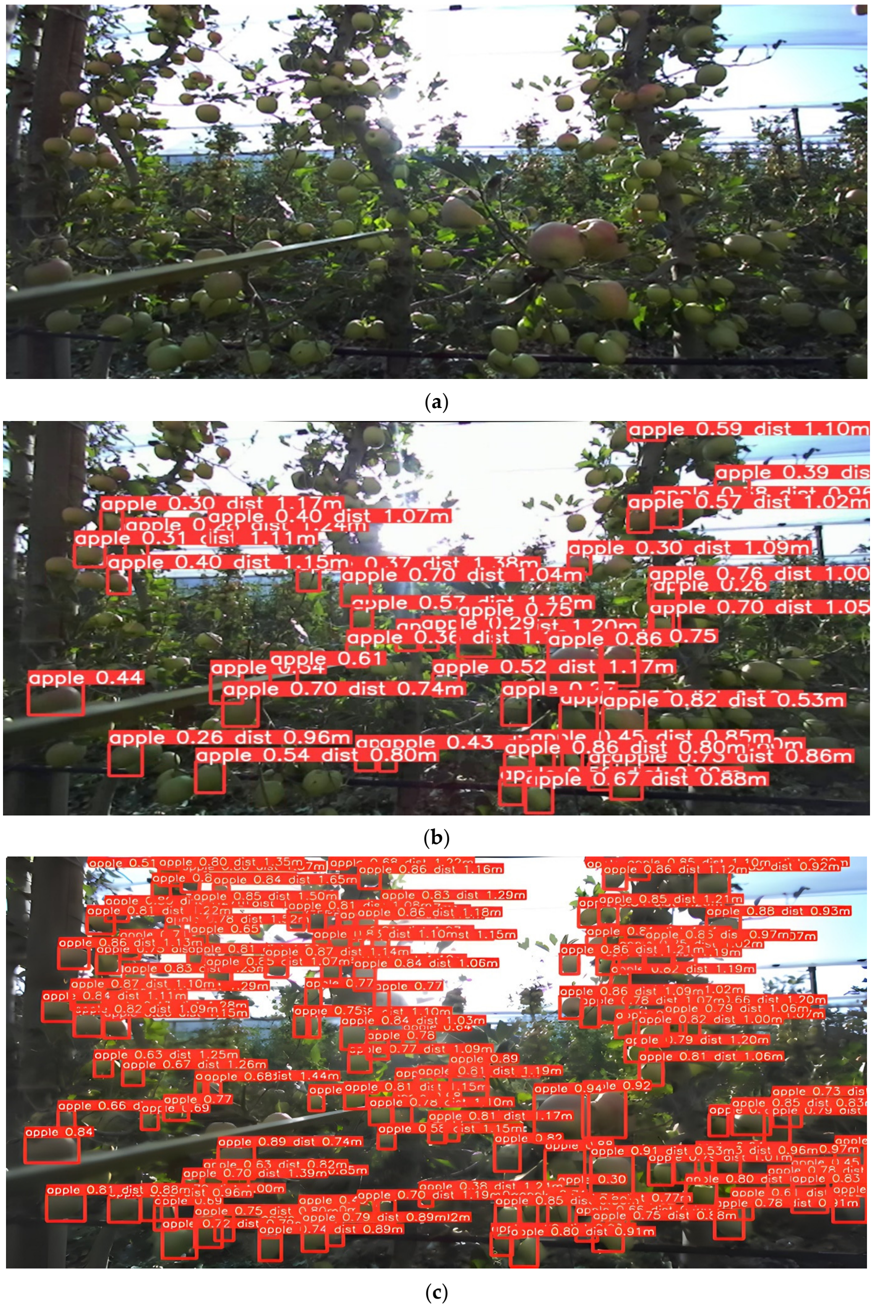
Figure 9. (**a**) Performance comparison bar graph of original images with existing models; (**b**) Performance comparison bar graph of illumination variation images with existing models.

3.3. Performance of Apple Tracking

To evaluate the effectiveness of the multi-object tracking methods, we considered the DeepSORT method and proposed the ByteTrack method for tracking and counting the apples [50,51]. Regarding multi-object tracking, the approach suggested in this study used deep learning. Therefore, it may be considered as an identical benchmark. In addition, the effectiveness of multi-object tracking was evaluated by directly using the trained YOLOv7 + MAM model for video detection. For the tracking and counting studies, three apple videos were chosen, and the following Equation (5) for mean absolute percent error (MAPE) was applied to compare the counting accuracy of the automated system with the manually recorded results:

$$\text{MAPE} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{X_t^i - Y_t^i}{Y_t^i}\right| \tag{5}$$

where Y_t^i indicates the entire number of manually counted apples in the collected video sequence, X_t^i shows the results of the apples counting process utilizing two multi-objective tracking algorithms, n is the total number of videos that need to be recognized, and i represents the current initial video. The choice of this indicator makes it feasible to observe the general characteristics of the model visually. Table 2 presents the comparative results of apple tracking and counting results using the methods DeepSORT and ByteTrack. The three apple videos considered were apple video ID1, apple video ID2, and apple video ID3. The video lengths of live inference of apples were 2.51 min, 1.48 min, and 0.41 s, respectively. After video detection with the proposed model, the detected results were forwarded to tracking using the ByteTrack algorithm. We employed the ByteTrack implementation with the developed YOLOv7 + MAM detection model. In this proposed execution, a tracker class was initiated, and appropriate tracks of the tracker instances were updated for every image in the video stream using the detections. Two formats were available for entering the detection: [h1, g1, h2, g2, score] or [h1, g1, h2, g2, object_score, class_score, class_id]. A set of active tracklets with attributes such as track_id, present frame bounding box, and confidence score may be found in the output online targets. The performance comparison is illustrated in Figure 12a,b with respect to the apple counting applied by the DeepSORT and ByteTrack techniques.For unblemished transparency of the tracking count of apples, only the count of tracking ID for each apple was displayed in the output, as shown in Figure 13a–c. The output also displayed the count of the number of apples being tracked in the left top corner of the video stream considered for the experiments at different intervals of time.



(a)

(b)

(c)

Figure 10. (**a**) Drone-centered image for apple detection in an apple orchard; (**b**) Inference of apple detection in an apple orchard using YOLOv5 + MAM; (**c**) Inference of apple detection in an orchard using proposed YOLOv7 + MAM.

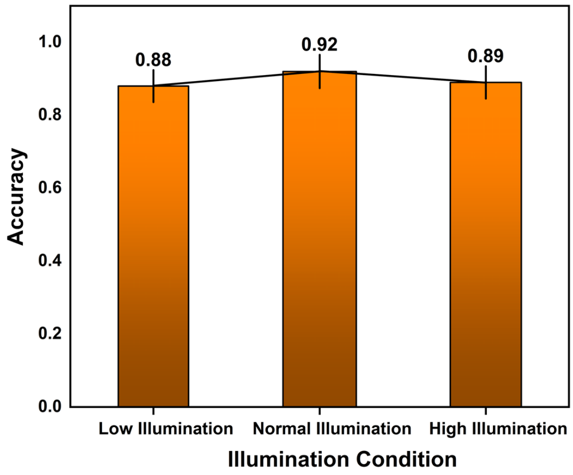


Figure 11. Model's accuracy under various lighting conditions.

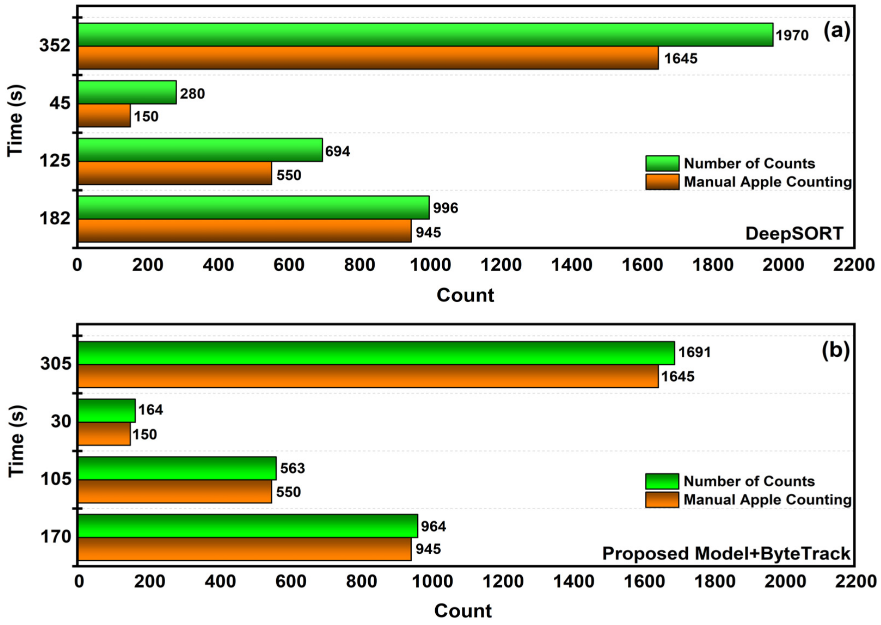
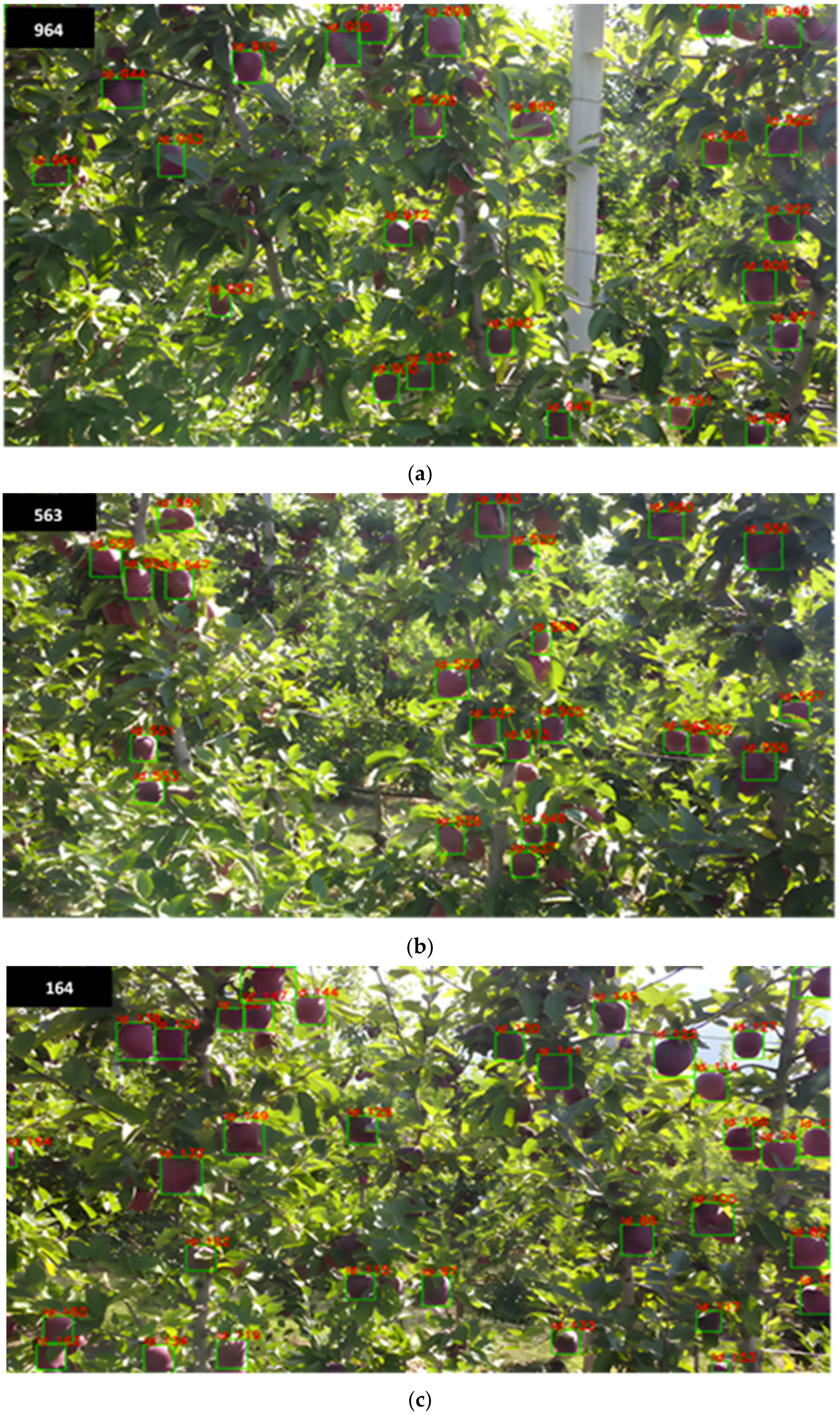


Figure 12. (**a**) Performance comparison graph of manual counting and DeepSORT tracking technique; (**b**) Performance comparison graph of manual counting and proposed tracking technique.

Table 2. Obtained experimental MAPE results of counting apples in live video inference.

Apple Video ID	Manual Apple Counting	Tracking Methods Employed after YOLOv7 + MAM	Count Time in Seconds	Number of Counts	MAPE
1	945	DeepSORT	182	996	0.053
		Proposed Model + ByteTrack	170	964	0.026
2	550	DeepSORT	125	694	0.261
		Proposed Model + ByteTrack	105	563	0.023
3	150	DeepSORT	45	280	0.866
		Proposed Model + ByteTrack	30	164	0.093
All	1645	DeepSORT	352	1970	0.197
		Proposed Model + ByteTrack	305	1691	0.027



(**a**)

(**b**)

(**c**)

Figure 13. (**a**) The apple tracking results of Video ID1 after applyingYOLOv7 + MAM + ByteTrack; (**b**) The apple tracking results of Video ID2 after applying YOLOv7 + MAM + ByteTrack; (**c**) The apple tracking results of Video ID3 after applying YOLOv7 + MAM + ByteTrack.

4. Discussion

Based on the results mentioned in Table 1, we observed that YOLOv7+ MAM demonstrated a significant increase in overall performance along with enhancements in precision, recall, and F1 score. The results presented here imply that including an attention mechanism improves the capacity to recognize apples using YOLOv5 and YOLOv7. Interestingly, adding the multi-head attention mechanism did not significantly alter the model's size impact on running speed. YOLOv7, nevertheless, performed better overall compared to the YOLOv5 network. Deep neural networks can combine fruit graphs with various feature distributions to enhance overall generalization performance in light of migration learning. As a result, the prediction model may be built using several factors discovered by localized migration learning. This study revealed that the multi-head attention mechanism's inclusion had no appreciable impact on the model's size scale. This could be because the attention mechanism additional layer deepened the model's understanding of small objects while adding little to the overall complexity of the computation. As a result, the model's size remained relatively high. This could facilitate model deployment by allowing the compression to concentrate primarily on the optimization aspect of backbone network pruning rather than the structure's overall compression [52]. The primary benefit of this method is labeling the apple center and not the bounding box. This is quite beneficial in dense orchards. In addition to using cutting-edge technologies, YOLOv5 performed more accurately than previous models [53]. However, low light, motion blur, and complex backgrounds can affect how well these systems operate. We developed a novel deep learning mechanism based on YOLOv7 and a multi-head attention mechanism to address these issues. To address size changes, we implemented an attribute extension model operation on top of this architecture. In distinguishing between mature and immature apples, size is a critical aspect. Mature apples have larger diameters than immature ones. During training, the model learned to distinguish ripe apples based on the average size of their bounding boxes in the training sample. Therefore, the model implicitly learned a limit or range of permissible bounding box sizes for mature apples. When the model was applied for inference on live videos, it used the previously learned criteria to assess if a detected apple was likely to be a mature apple. If the bounding box associated with an object was inside the learned threshold for matured apples, the model considered it a valid detection. However, if the bounding box size was less than this criterion, signifying that the apple detected was most likely an immature apple, the model did not consider it a valid detection. Determining the three-dimensional location of every identified apple in the environment is essential for apple detection, and here is where depth estimation comes into play. This may be used to determine the distance between each apple and the camera. The YOLOv7 bounding boxes and depth information were linked to provide the three-dimensional location data (x, y, z) for every detected apple. ByteTrack used the identified apples' 3D coordinates as the starting point for tracking objects.

The counting results depicted that our suggested YOLOv7 + MAM + ByteTrack tracking approach outperformed the DeepSORT tracking method. Given that the suggested machine learning model produced the least amount of error, it was regarded as an efficient model and had the lowest MAPE [54,55]. The proposed model tracking experiment was performed on three different videos, and the consolidated MAPE applying DeepSORT was 0.197 and our proposed model attained a low MAPE of 0.027. The DeepSORT [56] technique also provided almost near results, but the duplication of apples and background apples that were not measured for the series of sequence counts made the model vulnerable. However, ByteTrack along with the recommended detection method categorized the apples in the foreground and background and included only the targeted apples in the count. When bounding boxes of apples were recognized, DeepSORT employed the ReID identifying model to link them across frames. If an apple could not be connected, SORT used the Kalman filter's predicted bounding box movements to link it between frames. It included only the bounding boxes with relatively high confidence. On the contrary, ByteTrack tracked the apples between frames solely by predicting their movements, using bounding boxes that were computed using the Kalman filter, eliminating the need for ReID. As a result, it shared technical similarities with DeepSort's Sort process. Nevertheless, dividing the processing into two stages, the first procedure aimed at the boundaries of the boxes having high confidence values and the second one with low confidence values, enhanced the performance.

To enhance the tracker performance, specific hyper parameters were utilized, such as MIN_THRESHOLD, which was set to 0.001 to retrieve nearly all detections. Bounding boxes, which were regarded as background boxes, were further filtered in the current model by a hard-coded background threshold set at 0.1. We could adjust MIN_THRESHOLD to values greater than 0.1 if we require more precision in our detection. However, this could exclude critical occluded object detection. To determine whether the threshold chosen offers the right quality, we should qualitatively review the situation.

It should also be emphasized that counting apples and recognizing their positions are two independent problems in the context of apple detection and tracking. The detection and tracking approaches also require different algorithms. The detection phase refers to the physical coordinates or bounding box of each apple within an image or video frame. This challenge requires recognizing the existence of apples and precisely determining their locations. From a technical aspect, detecting positions demands the creation of object identification algorithms capable of not only identifying objects but also providing precise spatial localization data. In terms of detecting apple positions, the algorithm's purpose is to provide bounding box coordinates for each discovered apple. These coordinates define the exact geographical location and size of each apple in the scene.

The counting refers to determining the total number of apples in a given scene or image. This assignment often entails identifying each apple and then adding it to the count of the total. The ID generated for each individual bounding box for each apple is not repeated. Each object is recognized with a unique ID in the live video captured by drone. The primary goal of the counting apples algorithm is to identify whether each observed apple object correlates precisely with the incremented ID number and the number of apples counted in the scene.

The comparative analysis illustrates that the proposed integrated approach achieved optimum performance even under different lightening conditions. Table 3 shows the time complexity for the YOLOv5 and YOLOv7 models combined with DeepSORT and ByteTrack at an mAP of 0.5 accuracy, including CPU and GPU time. The proposed approach resulted in the best accuracy, with low CPU and GPU time.

Table 3. Analyzing the performance of time complexity of tested models.

Models	Parameters (Million)	Accuracy (mAP 0.5)	CPU Time (ms)	GPU Time (ms)
YOLOv5 + MAM + DeepSORT	32.5	75.20	320	11.3
YOLOv7 + MAM + DeepSORT	24.6	79.32	220	9.1
YOLOv5 + MAM + ByteTrack	17.3	83.55	161	8.2
YOLOv7 + MAM + ByteTrack	11.5	92.35	71	6.4

To summarize the discussion, we found that the YOLOv7 + MAM detection head in conjunction with the ByteTrack tracking algorithm produced the best experimental results. The essential parameters for this approach are provided in Section 3. These findings suggest that a multi-head attention mechanism can enhance the detector's performance and that processing images with ByteTrack can improve its efficacy in multi-object tracking. These findings also provide a better framework for fruit counting research in the future. There are a few factors that should be considered, like deployment of GPU in different farm fields of apples that vary based on different geographical and growth environment conditions. The research is low cost and the proposed systems are scalable, allowing monitoring of orchards of all sizes, from small family farms to large commercial enterprises. Whether tracking acres or thousands of hectares, the proposed model offers a versatile and scalable alternative for apple detection, depth estimate, and agricultural monitoring. The client could obtain an exact count of the apples ready for harvest. Although color, shape, and size are minimal factors as limitations, the speed of the drone and the detection and tracking rate are major concerns that might affect the overall present fruit counting methodology. In the future, we plan to integrate different attention mechanisms with the latest lightweight models to attain a balance between speed and performance in fruit counting.

5. Conclusions

In this work, we proposed a YOLOv7 framework with a multi-head attention mechanism integrated with a ByteTrack multi-object tracking system to detect and count apple fruits in orchards. The results of our experiments demonstrated that the accuracy and robustness of apple identification were significantly improved by combining YOLOv7 with a multi-head attention mechanism. Even with difficult lighting circumstances and a variety of apple orientations, the model could successfully detect the apples along with the depth estimation of each apple, which enabled determination of the distance between each apple and the drone camera. Furthermore, ByteTrack for apple counting ensured our system's effectiveness. Apple counting was made simple and quick by seamlessly integrating ByteTrack for apple detection and tracking. ByteTrack ensured that tracking continued even if the apples moved, varied in appearance, or momentarily disappeared from the drone's vision. The method successfully dealt with occlusions and various sizes of apples in the orchard, which helped to provide precise and accurate counting outcomes. To verify the efficacy of our suggested model, we carried out comprehensive comparison tests with many current detection and counting techniques. The outcomes demonstrated how well the model performed in achieving the ideal balance between speed and precision, which makes it an invaluable tool for precision agriculture. We believe that our work paves the way for more developments in agricultural automation and establishes a solid basis for future research on object recognition and counting in complicated contexts.

Author Contributions: Conceptualization, P.K.S., F.M. and R.R.; methodology, P.K.S., F.M., R.R. and R.d.A.S.; software, R.R., R.d.A.S. and P.K.S.; formal analysis, P.K.S., F.M., J.M. and J.M.J.; resources, F.M., J.M.J. and J.M.; data curation, P.K.S. and R.R.; writing—original draft preparation, P.K.S. and F.M.; writing—review and editing, P.K.S., R.R., R.d.A.S. and F.M.; supervision, F.M., J.M.J. and J.M.; project administration, F.M., J.M.J. and J.M.; funding acquisition, F.M. and J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fondazione Caritro (Trento, Italy) under the program 'Bando Post-doc 2021'. International collaboration is co-funded by the Italian Ministry of Foreign Affairs and International Cooperation and the Brazilian National Council of State Funding Agencies.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This research was carried out within the framework of a project entitled 'System based on Artificial Intelligence and Drones for Apple Picking Automation in Trentino' funded by the Fondazione Caritro (Trento, Italy). International collaboration was possible thanks to the project entitled "Deep Learning for Precision Farming Mapping", co-founded by the Italian Ministry of Foreign Affairs and International Cooperation and the Brazilian National Council of State Funding Agencies. We would also like to acknowledge the use of Grammarly to improve the clarity and correctness of the English language in the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055. [CrossRef]
2. Murala, S.; Vipparthi, S.K.; Akhtar, Z. Vision Based Computing Systems for Healthcare Applications. *J. Healthc. Eng.* **2019**, *2019*, 9581275. [CrossRef] [PubMed]
3. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection during Different Growth Stages in Orchards Using the Improved YOLO-V3 Model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
4. Zhao, C. Current situations and prospects of smart agriculture. *J. South China Agric. Univ.* **2021**, *42*, 1–7.
5. Cohen, O.; Linker, R.; Naor, A. Estimation of the number of apples in color images recorded in orchards. In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, Nanchang, China, 22–25 October 2010; pp. 630–642.
6. Ji, W.; Zhao, D.; Cheng, F.; Xu, B.; Zhang, Y.; Wang, J. Automatic recognition vision system guided for apple harvesting robot. *Comput. Electr. Eng.* **2012**, *38*, 1186–1195. [CrossRef]
7. Bulanon, D.; Kataoka, T.; Zhang, S.; Ota, Y.; Hiroma, T. Optimal Thresholding for the Automatic Recognition of Apple Fruits. In Proceedings of the 2001 ASAE Annual Meeting, Sacramento, CA, USA, 29 July–1 August 2001. [CrossRef]
8. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
9. Prasetiyowati, M.I.; Maulidevi, N.U.; Surendro, K. Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. *J. Big Data* **2021**, *8*, 84. [CrossRef]
10. López-Morales, J.A.; Martínez, J.A.; Skarmeta, A.F. Digital transformation of agriculture through the use of an interoperable platform. *Sensors* **2020**, *20*, 1153. [CrossRef]
11. Sun, J.; He, X.; Ge, X.; Wu, X.; Shen, J.; Song, Y. Detection of Key Organs in Tomato Based on Deep Migration Learning in a Complex Background. *Agriculture* **2018**, *8*, 196. [CrossRef]
12. Bulanon, D.; Kataoka, T. Fruit detection system and an end effector for robotic harvesting of Fuji apples. *Agric. Eng. Int. CIGR E-J.* **2010**, *12*, 203–210.
13. Tian, Y.; Duan, H.; Luo, R.; Zhang, Y.; Jia, W.; Lian, J.; Zheng, Y.; Ruan, C.; Li, C. Fast Recognition and Location of Target Fruit Based on Depth Information. *IEEE Access* **2019**, *7*, 170553–170563. [CrossRef]
14. Hu, L. An Improved YOLOv5 Algorithm of Target Recognition. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023. [CrossRef]
15. Wang, J.; Su, Y.; Yao, J.; Liu, M.; Du, Y.; Wu, X.; Huang, L.; Zhao, M. Apple rapid recognition and processing method based on an improved version of YOLOv5. *Ecol. Inform.* **2023**, *77*, 102196. [CrossRef]
16. Shang, Y.; Xu, X.; Jiao, Y.; Wang, Z.; Hua, Z.; Song, H. Using lightweight deep learning algorithm for real-time detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2023**, *207*, 107765. [CrossRef]
17. Mirbod, O.; Choi, D.; Heinemann, P.H.; Marini, R.P.; He, L. On-tree apple fruit size estimation using stereo vision with deep learning-based occlusion handling. *Biosyst. Eng.* **2023**, *226*, 27–42. [CrossRef]
18. Gené-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Escolà, A.; Gregorio, E. PFuji-Size dataset: A collection of images and photogrammetry-derived 3D point clouds with ground truth annotations for Fuji apple detection and size estimation in field conditions. *Data Brief* **2021**, *39*, 107629. [CrossRef] [PubMed]
19. Biffi, L.J.; Mitishita, E.; Liesenberg, V.; Santos, A.A.; Gonçalves, D.N.; Estrabis, N.V.; Silva, J.D.; Osco, L.P.; Ramos, A.P.; Centeno, J.A.; et al. ATSS Deep Learning-Based Approach to Detect Apple Fruits. *Remote Sens.* **2020**, *13*, 54. [CrossRef]
20. Ma, L.; Zhao, L.; Wang, Z.; Zhang, J.; Chen, G. Detection and Counting of Small Target Apples under Complicated Environments by Using Improved YOLOv7-tiny. *Agronomy* **2023**, *13*, 1419. [CrossRef]
21. Chen, J.; Mai, H.; Luo, L.; Chen, X.; Wu, K. Effective Feature Fusion Network in BIFPN for Small Object Detection. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 699–703.
22. Hodson, T.O. Root-Mean-Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not. *Geosci. Model Dev.* **2022**, *15*, 5481–5487. [CrossRef]
23. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsboui, T. Domain feature mapping with YOLOv7 for automated edge-based pallet racking inspections. *Sensors* **2022**, *22*, 6927. [CrossRef] [PubMed]
24. Wang, J.L.; Li, A.Y.; Huang, M.; Ibrahim, A.K.; Zhuang, H.; Ali, A.M. Classification of white blood cells with pattern net-fused ensemble of convolutional neural networks (pecnn). In Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 6–8 December 2018; pp. 325–330.
25. Brock, H.; Rengot, J.; Nakadai, K. Augmenting sparse corpora for enhanced sign language recognition and generation. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018) and the 8th Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, Miyazaki, Japan, 7–12 May 2018; pp. 7–12.
26. Yang, H.; Liu, Y.; Wang, S.; Qu, H.; Li, N.; Wu, J.; Yan, Y.; Zhang, H.; Wang, J.; Qiu, J. Improved Apple Fruit Target Recognition Method Based on YOLOv7 Model. *Agriculture* **2023**, *13*, 1278. [CrossRef]

27. Shindo, T.; Watanabe, T.; Yamada, K.; Watanabe, H. Accuracy improvement of object detection in VVC coded video using YOLO-v7 features. *arXiv* **2023**, arXiv:2304.00689v1.
28. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
29. Hu, J.; Fan, C.; Wang, Z.; Ruan, J.; Wu, S. Fruit Detection and Counting in Apple Orchards Based on Improved Yolov7 and Multi-Object Tracking Methods. *Sensors* **2023**, *23*, 5903. [CrossRef] [PubMed]
30. Xiao, B.; Nguyen, M.; Yan, W.Q. Apple ripeness identification from digital images using transformers. *Multimedia Tools Appl.* **2023**, *83*, 7811–7825. [CrossRef]
31. Chen, X.; Pu, H.; He, Y.; Lai, M.; Zhang, D.; Chen, J.; Pu, H. An Efficient Method for Monitoring Birds Based on Object Detection and Multi-Object Tracking Networks. *Animals* **2023**, *13*, 1713. [CrossRef] [PubMed]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
33. Thakuria, A.; Erkinbaev, C. Improving the network architecture of YOLOv7 to achieve real-time grading of canola based on kernel health. *Smart Agric. Technol.* **2023**, *5*, 100300. [CrossRef]
34. Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V. Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415. *Symmetry* **2022**, *14*, 148. [CrossRef]
35. Stereolabs Docs: API Reference, Tutorials, and Integration. Available online: https://docs.stereolabs.com/depth-sensing/depth-settings (accessed on 5 December 2023).
36. Wang, H.; Feng, J.; Yin, H. Improved Method for Apple Fruit Target Detection Based on YOLOv5s. *Agriculture* **2023**, *13*, 2167. [CrossRef]
37. Zhao, Z.; Wang, J.; Zhao, H. Research on Apple Recognition Algorithm in Complex Orchard Environment Based on Deep Learning. *Sensors* **2023**, *23*, 5425. [CrossRef]
38. Kumar, S.P.; Naveen Kumar, K. Drone-based apple detection: Finding the depth of apples using YOLOv7 architecture with multi-head attention mechanism. *Smart Agric. Technol.* **2023**, *5*, 100311. [CrossRef]
39. Liu, J.; Wang, C.; Xing, J. YOLOv5-ACS: Improved Model for Apple Detection and Positioning in Apple Forests in Complex Scenes. *Forests* **2023**, *14*, 2304. [CrossRef]
40. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516. [CrossRef]
41. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. Foveabox: Beyound anchor-based object detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [CrossRef]
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef]
43. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
44. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Track: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
45. Yu, C.; Feng, Z.; Wu, Z.; Wei, R.; Song, B.; Cao, C. HB-YOLO: An Improved YOLOv7 Algorithm for Dim-Object Tracking in Satellite Remote Sensing Videos. *Remote Sens.* **2023**, *15*, 3551. [CrossRef]
46. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina at chapel hill: Chapel hill, NC, USA, 1995; p. 2.
47. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist.* **2005**, *52*, 7–21. [CrossRef]
48. Yang, H.; Chang, F.; Huang, Y.; Xu, M.; Zhao, Y.; Ma, L.; Su, H. Multi-object tracking using deep SORT and modified CenterNet in cotton seedling counting. *Comput. Electron. Agric.* **2022**, *202*, 107339. [CrossRef]
49. Fischer, T.; Huang, T.E.; Pang, J.; Qiu, L.; Chen, H.; Darrell, T.; Yu, F. QDTrack: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking. *arXiv* **2022**, arXiv:2210.06984. [CrossRef] [PubMed]
50. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-object Tracking by Associating Every Detection Box. In *Computer Vision—ECCV 2022*; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; ECCV 2022. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13682. [CrossRef]
51. Zheng, Z.; Li, J.; Qin, L. YOLO-BYTE: An efficient multi-object tracking algorithm for automatic monitoring of dairy cows. *Comput. Electron. Agric.* **2023**, *209*, 107857. [CrossRef]
52. Gennari, M.; Fawcett, R.; Prisacariu, V.A. DSConv: Efficient Convolution Operator. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
53. van Meekeren, A.; Aghaei, M.; Dijkstra, K. Exploring the Effectiveness of Dataset Synthesis: An application of Apple Detection in Orchards. *arXiv* **2013**, arXiv:2306.11763.
54. Gené-Mola, J.; Ferrer-Ferrer, M.; Gregorio, E.; Blok, P.M.; Hemming, J.; Morros, J.-R.; Rosell-Polo, J.R.; Vilaplana, V.; Ruiz-Hidalgo, J. Looking behind occlusions: A study on amodal segmentation for robust on-tree apple fruit size estimation. *Comput. Electron. Agric.* **2023**, *209*, 107854. [CrossRef]

55. Ferrer-Ferrer, M.; Ruiz-Hidalgo, J.; Gregorio, E.; Vilaplana, V.; Morros, J.-R.; Gené-Mola, J. Simultaneous fruit detection and size estimation using multitask deep neural networks. *Biosyst. Eng.* **2023**, *233*, 63–75. [CrossRef]
56. Abeyrathna, R.M.R.D.; Nakaguchi, V.M.; Minn, A.; Ahamed, T. Recognition and Counting of Apples in a Dynamic State Using a 3D Camera and Deep Learning Algorithms for Robotic Harvesting Systems. *Sensors* **2023**, *23*, 3810. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.