*remote sensing*

MDPI

*Article*

# Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO

**Praveen Kumar Sekharamantry *, Farid Melgani and Jonni Malacarne**

Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy; farid.melgani@unitn.it (F.M.); jonni@bluetensor.ai (J.M.)

* Correspondence: pk.sekharamantry@unitn.it; Tel.: +39-389-009-3134

**Abstract:** Horticulture and agriculture are considered as the important pillars of any economy. Current technological advancements have led to the development of several new technologies which are useful in atomizing the agriculture process. Apple farming has a significant role in Italy's agriculture domain where manual labor is widely employed for apple picking which can be replaced by automated robot mechanisms. However, these mechanisms are based on computer vision methods. These methods focus on detection, localization and tracking the apple fruits in given video frames. Later, appropriate actions can be taken to enhance the production and harvesting. Several techniques have been presented for apple detection, but complex background, noise and image blurriness are the major causes which can deteriorate the performance of the system. Thus, in this work, we present a deep learning-based scheme to detect apples which uses Yolov5 architecture in live apple farm images. We further improve the Yolov5 architecture by incorporating an adaptive pooling scheme and attribute augmentation model. This model detects the smaller objects and improves the feature quality to detect the apples in complex backgrounds. Moreover, a loss function is also incorporated to obtain the accurate bounding box which helps to maximize the detection accuracy. The comparative study shows that the proposed approach with the improved Yolov5 architecture achieves overall accuracy of 0.97, 0.99, and 0.98 in terms of precision, recall, and F1-score, respectively.

**Keywords:** apple detection; localization; deep learning; YOLOv5; attention model



**Citation:** 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. https://doi.org/10.3390/rs15061516

Academic Editor: Thomas Alexandridis

Received: 30 January 2023
Revised: 2 March 2023
Accepted: 7 March 2023
Published: 9 March 2023



## 1. Introduction

Object detection is a basic but difficult task in computer vision, with the goal of categorizing and localizing each target in an image [1]. Computer vision-based systems have been used in a variety of applications recently, including biomedical [2], remote sensing, agricultural and farming monitoring, multimedia, etc. [3]. The goal of this study is to use a deep learning-based method for farm automation. Apples are one of the most frequently cultivated fruits, yet they are still harvested by hand, making robot harvesting desirable. Apples are high in nutrients, low in fat, high in carbohydrates, and contain vitamins C and E. Apples may also be grown in a range of conditions and have a great economic value. Italy is one of the world's greatest apple growers, with the largest planting area and fresh apple export [4]. Italy's yearly apple output was 2.4 million tons in 2021. Because of the high need for labor during the harvest season, automated picking robots are in high demand [5]. A vision system and an end-effector system are the two major subsystems of a picking robot [6]. By recognizing and localizing apples, the vision system directs the robot end-effector to pluck apples from trees. Thus, detection and tracking the apples becomes an important task for these applications [7].

Moreover, the accurate detection of apples can help to predict the production quantity, maturity as well as schedule recreational activities such as fruit picking, disaster damage, and employing the robots in agricultural automation. This process is based on the fruit detection and tracking which is performed on the sequence of image frames. Detection or

localization of apple fruit in the image is an important task for these applications. Currently, several techniques have been presented to locate the objects. These traditional methods use binocular localization [8], RGB-D cameras [9] and laser scanner [10] techniques. However, the performance of these methods is limited due to disturbance of ambient light which causes the phase error resulting in measurement error, measurement limitations and effect of external environment which produces errors in measurement. On the other hand, image processing-based schemes such as graph-cuts [11], color segmentation [12], SIFT (Scale Invariant Feature Transform) [13], etc., are also adopted widely but the accuracy of these systems is very poor in occlusion, noise, and illuminated images.

Currently, the demand of machine learning and deep learning-based methods has increased drastically. Moreover, these techniques are widely adopted in agriculture-related applications such as crop segmentation, crop detection, yield analysis, etc. Based on the concept of CNN, the authors in [14] used the CNN model to obtain the desired region in the image and applied successive CNN approach to count the fruits. In [15], Dias et al. presented a combined CNN and SVM model to handle the background complexity issue. Similarly, Faster RCNN architecture was employed with region proposal networks to obtain the accurate region of interest (ROI). Faster RCNN with VGG16 is considered a promising technique for fruit detection. Faster R-CNN, on the other hand, is made up of region proposal networks (RPN) and classification networks that achieved outstanding accuracy results, but time complexity remains a challenging task, making it unable to obtain decent results in real-time with high image resolution. The You Only Look Once (YOLO) approach [16] is a regression problem that deals with classification and localization. Without RPN, a YOLO network conducts regression directly to recognize targets in an image, making it quick and suitable for real-time applications. The most recent versions (YOLOv3 and YOLOv5 [17]) have great detection accuracy and speed, as well as the ability to detect tiny objects. Due to its complicated design, which needs greater processing power, the YOLOv3 model is not suited for real-time applications such as harvesting robots. The model parameters must be optimized to decrease computing complexity, which is required for deployment on edge devices such as the Jetson and Raspberry Pi. Biffi et al. [18] introduced the Adaptive Training Sample Selection (ATSS) deep learning scheme for apple fruit detection and applied this approach for close-range and low-cost terrestrial RGB images. The main advantage of this method is that it only labels the center point of the apple rather than the bounding box which shows a significant advantage in a heavily dense fruit orchard. The state-of-art techniques have reported that Yolov5 achieves better accuracy when compared with other models. However, the performance of these systems is degraded because of background complexity, motion blurriness, low illumination, etc. To overcome these issues, we present a new deep learning approach which is based on the YoloV5 model. Along with this architecture, we have incorporated attribute augmentation model and adaptive pooling operations to handle the size variations.

The main novelties of proposed approach are listed below:

- Feature generation plays an important role in apple detention tasks because of irregular size, occlusion and position variations. To consider this, we adopted the concept of feature pyramid network (FPN). However, the tradition FPN-based model suffers from the contextual information loss, therefore, we incorporated an attribute augmentation model which helps to mitigate the issue of contextual information loss and a feature enhancement model which improves the feature representation to increase the inference speed.
- In order to increase the robustness of the proposed approach, we apply a data augmentation scheme which includes several tasks such as brightness variation, image mirroring, rotation, motion blur, and adding noise.

The rest of the article is organized as follows: Section 2 describes the proposed solution which describes data acquisitions, data augmentation, a brief discussion about existing Yolo architectures and a description of the proposed architecture along with its components is presented. Section 3 presents the experimental discussion of the proposed approach and

comparative analysis to show the robustness of the proposed approach when compared with the traditional object detection schemes. Finally, Section 4 presents the concluding remarks and future scope of this technique for apple detection.

## 2. Proposed Model

### *2.1. Data Acquisition*

The data were acquired both by reflex cameras and by drone on two separate fields provided by two farmers. The fields are located at first part of Val di Non and there are two distinct varieties, red delicious and golden delicious. The photos and films were taken at fixed distances of 30 and 60 cm from the plant. The drone used is the popular model known as the DJI Mavic 3 [19,20]. The drone is equipped with tools for precision agriculture and is a powerful flagship camera drone. It is furnished with a Hasselblad 4/3 CMOS camera to facilitate professional-grade imaging. It also offers omnidirectional obstacle detection for an even flying experience with a determined flight range. The onboard sensors stay safe and in place in case there are abrupt and hard motor vibrations, it also features a fixed wing system.

The data collected by the drone was collected during a day of varying weather conditions in September. It was flown with no additional lights or artificial lighting. The drone's configuration was processed through the rtmp protocol, which is used to connect the camera to the drone's backend storage. Its maximum transmission distance is 80 m, and its height is 50 m without interference.

### *2.2. Data Augmentation*

During the aforementioned process, we observed that the distance between camera and trees varies, because some apple images are quite small whereas some images are bigger in size. According to the Figure 1, there is huge imbalance in the original data, moreover, real-time applications suffer from this type of uncertainty of input data, thus, training this type of imbalanced data may lead to issues of overfitting and it may degrade the detection accuracy. Similarly, the apple image acquisition model suffers from these challenges during image capturing. Thus, data augmentation becomes a prominent task in these types of computer vision applications where the size of objects varies frequently.

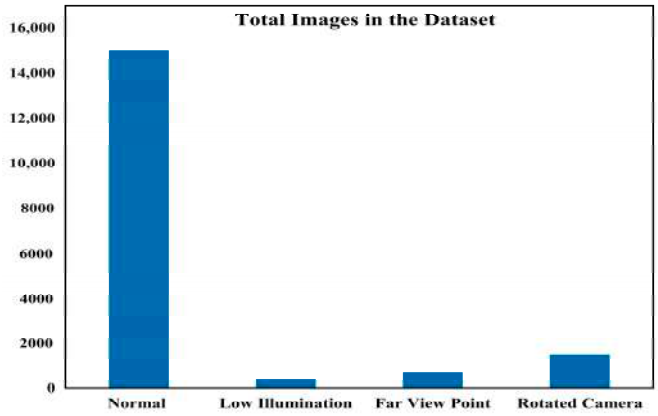



**Figure 1.** Captured apple images at varied conditions.

In this work, we have considered brightness enhancement, rotation, brightness reduction, Gaussian noise, and motion blur effects for data augmentation. The given Figure 2 shows the sample images of data augmentation. We have utilized the following models for image augmentation:

- Reducing and increasing the brightness of image: first, the image is converted into HSV space by using the 'rgb2hsv' function, in the next stage, the bright component, i.e., V is multiplied by different coefficients, the obtained HSV image is transformed

into RGB space by applying the 'hsv2rgb' function, resulting in a brightness enhancement. The brightness enhancement is carried out based on two intensity values $(H + S + 1.2V)$ and $(H + S + 1.6V)$, similarly, brightness reduction is carried out based on two intensity values as $(H + S + 0.6V)$ and $(H + S + 0.8V)$. This method increases and decreases the brightness which helps to learn the patterns in such a way that the model can detect the apples in conditions with poor illumination.

- Image mirroring: this is performed by mirroring the horizontal and vertical pixels. The horizontal mirroring is obtained by transforming the left and right side of image centering on the vertical center line of image. Similarly, the upper and lower sides are transformed on the horizontal centerline of image to generate the vertical mirroring. This scenario helps to obtain the augmented data and mirror images carry the same characteristics of an apple that we would want an image classifier to learn. Especially when involving tasks where the perspective of the image is unknown.
- Image rotation: in this augmentation, the image is rotated by 90°, 180°, and 270°. These rotations help to obtain the accurate detection of apples irrespective of image capturing angle. If the camera position is not fixed relative to objects, random rotation is likely a helpful image augmentation. Therefore, we consider this an augmentation task.
- Motion blur: the speed of the capturing device affects the quality of image capturing. Thus, in this stage, we include four types of motion blur for data augmentation. These motion filters are obtained by applying (6, 30), (6, −30), (7, 45) and (7, −45) motion blurs, respectively. Here, the motion blur is represented as $(\text{Len}, \theta)$Len is the length which represents the pixels of linear motion of camera, and θ is the angular degree in counterclockwise. Researchers suspect blur particularly obscures convolution's ability to locate edges in early levels of feature abstraction, causing inaccurate feature abstraction early in a network's training. Therefore, training the model with blurred data helps to obtain a better level of detection.
- Noisy image: in this process, we add Gaussian noise with variance of 0.02 to obtain the augmented data. Authors in [21,22] used Gaussian noise for data augmentation where the variance of Gaussian noise is considered as 0.02. Moreover, characteristics of Gaussian noise make it more suitable to adopt it for experiments. Gaussian noise is caused due to sensor noise by poor illumination, high temperature and electronic circuit noise which can occur while capturing the image. In [16], the authors suggested that Gaussian noise represents the characteristics of human motion.

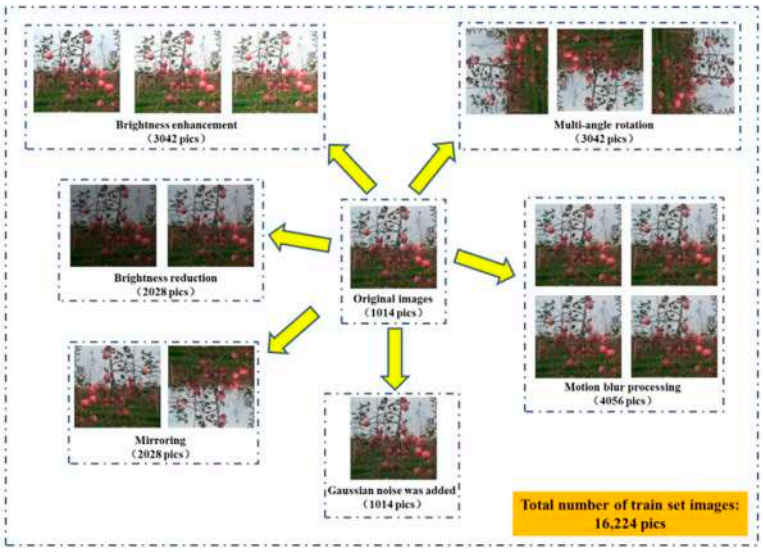



**Figure 2.** Data augmentation process.

The final training sets consist of 20,000 images used for training of apple targets recognition model, including 15,210 enhanced images and 1014 raw images. This dataset

contains different categories of images such as normal images, low illumination images, far viewpoint images, and rotated camera images where 15,000 images, 400 images, 800 images, and 1200 images belong to each variation. The given Figure 2 depicts the images considered after each augmentation process.

### *2.3. Yolov5 and Proposed Yolov5 Architecture*

YOLO is based on the neural network scheme and is widely adopted for real-time object detection. The accuracy and speed of the approach are the substantial parameters which make it one of the most popular approaches. Several architectures have been introduced in this Yolo series such as Yolov3, Yolov4, Yolov5, etc. The basic architectures of Yolov4 and Yolov5 are depicted in the given Figure 3.

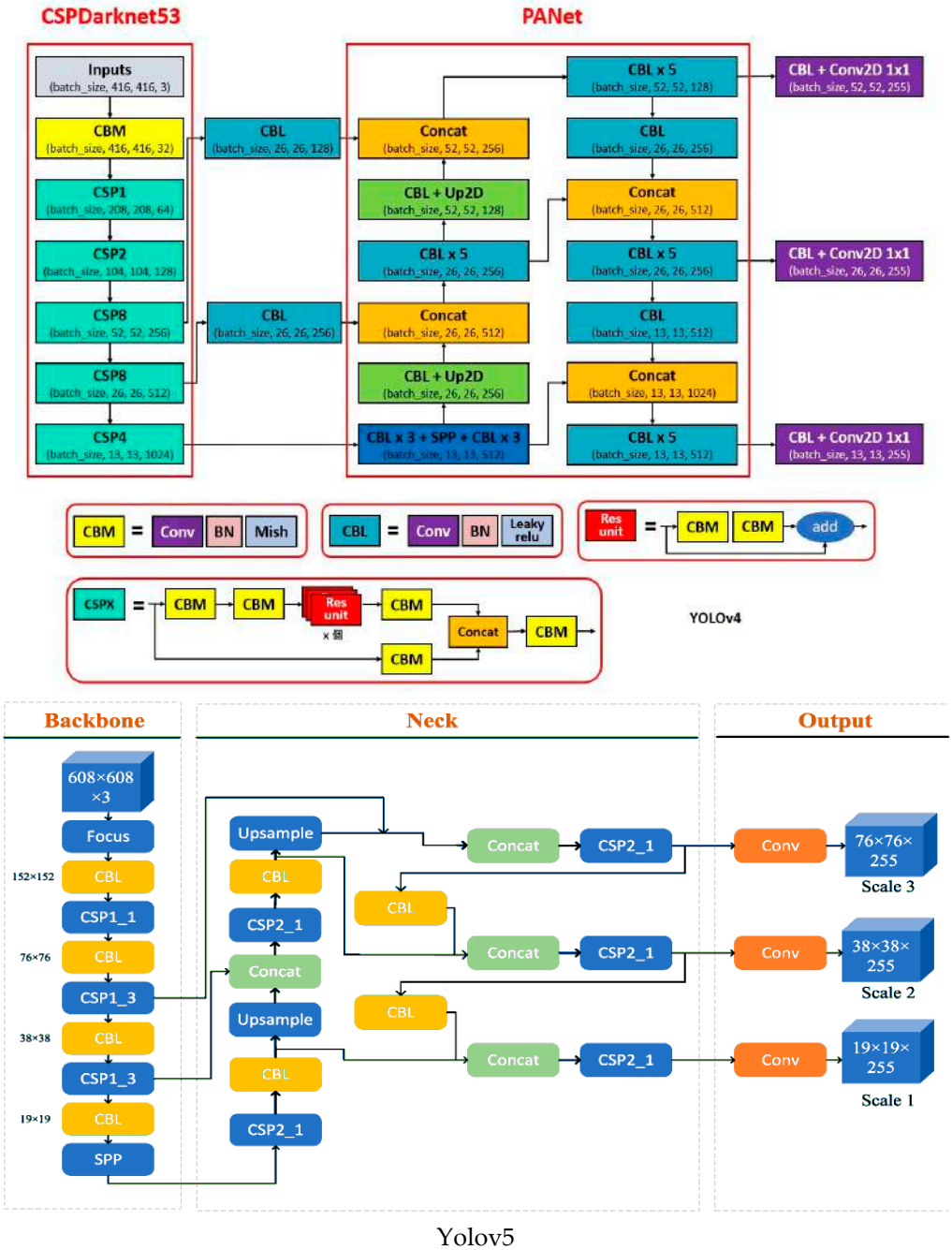



**Figure 3.** Yolov4 and Yolov5 architectures.

The traditional yolo architectures suffer from several performance issues such as performance of Yolov1 degrades based upon the closeness of object. Recently, the new architecture such as Yolov3 and Yolov4 [23] have been introduced to overcome the issues of object detection by incorporating batch normalization, high resolution classifiers, multi scale training which helps to improve the mean average precision (mAP), recall and precision. Yolov5 is the latest architecture of the YOLO series which provides improved accuracy with faster computation. Moreover, the weight file size of Yolov5 is 90% smaller than the Yolov4 which makes it more suitable to implement for real-time devices. We adopt this model for apple detection; however, we have modified the architecture of Yolov5 by introducing attention-based network and a novel attribute augmentation model. Initially, the attention mechanism was used in machine translation application where it was applied to guide deep neural network modules by facilitating the focus point and highlighting the important attributes and minimizing less significant attributes [24]. Recently, attention-based mechanisms have been widely adopted in deep learning applications such as remote sensing applications. Specifically, the attention mechanism gives more weightage to relevant parts and minimizes the irrelevant parts by assigning lower weightage to them [25].

Generally, when the color and shape of objects are different from the background, i.e., when the background and foreground are different, the attention mechanism has been proved as promising technique to improve the accuracy. For example, the authors in [26] have adopted the attention mechanism to extract the foreground features for human detection via attention mechanism. Similarly, in the proposed model, the shape and color of apples are different from the foreground and background. The traditional YOLO architectures fail [27] to maintain the context information throughout the network, whereas the proposed attention module mitigates this loss of contextual information which improves the learning process of the network. Similarly, the attribute augmentation model helps to map the low-level attributes to a high level feature.

The given Figure 4 depicts the proposed network architecture for apple detection based on the Yolov5 model. In this work, we have adopted Yolov5 as the base architecture and incorporated a proposed attention and feature augmentation mechanism. The complete architecture is divided into three modules: a backbone, neck and prediction module. The backbone architecture is a deep learning-based architecture which is used as feature extractor module. The backbone module is comprised of convolutional neural network which performs pooling of image pixels to generate the features at different levels of coarseness. This model is trained on the ImageNet classification dataset.

The next module is the neck module which is used for combining and mixing the ConvNet layer representations and passes this as input to the prediction module. This module has several options, such as feature pyramid network (FPN), path aggregation network (PAN), bi-directional feature pyramid network (BiFPN), and many more [28]. The basic architecture of these modules is presented in the given Figure 5.

This work mainly uses the PANet architecture because the traditional techniques used the top down FPN model. The performance of this model is limited by the one-way information flow. Therefore, PANet is considered as an innovative solution because it adds an additional bottom-up path aggregation network.

The backbone and neck modules contain Cross Stage Partial (CSP) modules. This process divides the input into two parts where one part of the input is evaluated through the block and other input is concatenated directly. This process of CSP helps to minimize the time requirement for processing. In this network, we have two types of CSPs as CSP1_X and CSP2_X. According to the CSP1_X, the input is divided into two parts where one part is processed through the CBL, Res Unit and Conv layer whereas the other part is directly processed through the convolution layer. Further, these two inputs are given to the concatenate block to perform the merge operation. Later, the merged output is processed through the Batch Normalization (BN), ReLU and CBL blocks. Similarly, CSP2_X is applied to the neck module of the network. In CSP2_X, the ResUnit block is replaced with the CBL

block. The CSP2_X increases the feature fusion process and overcomes the issue of gradient information repetition to obtain the better accuracy.

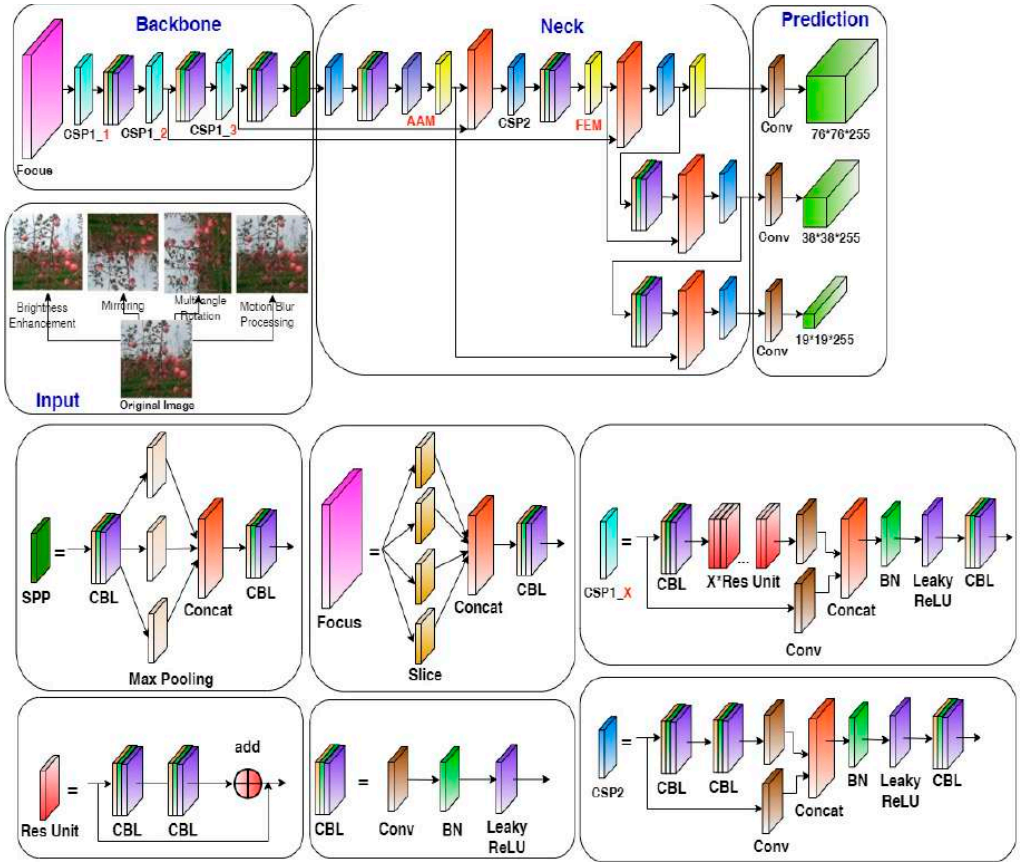



**Figure 4.** Proposed architecture.

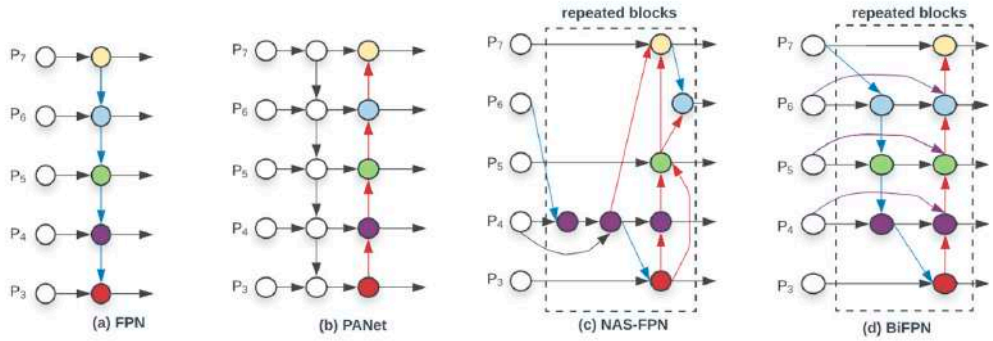



**Figure 5.** Common architecture used in YOLO neck modules.

Let us consider that feature maps of input image are represented as $\{C_1, C_2, C_3, C_4, C_5\}$ by processing through multiple convolutions. Here, we incorporate the attention module to $C_5$ to generate the feature map $M_6$ which is combined with $M_5$ and fused with other features at different levels. In order to fuse with other features, it is propagated towards downward with downsampling. On the other hand, the PANet uses upsampling to bridge the information path between the lower and topmost layers. The given Figure 6 depicts the combination of the attention module with the attribute augmentation module.

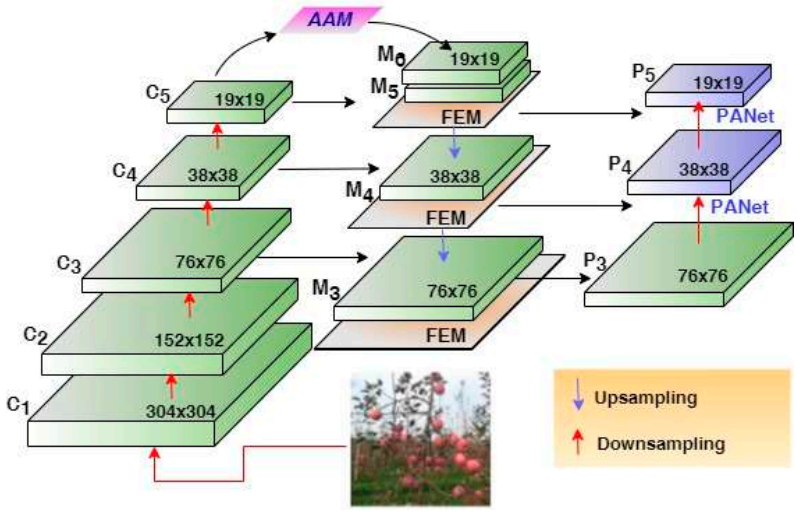



**Figure 6.** Combined attention with attribute augmentation model.

The proposed attention mechanism is performed in two steps: (a) obtaining multiple context attributes and (b) generation of weighted feature maps. The given Figure 7 depicts the architecture of proposed attention module with an adaptive pooling. This network contains an adaptive pooling layer with the pooling coefficient $\beta = [0.1, 0.5]$. These coefficients are varied according to the size of data. Moreover, it helps to obtain local and global characteristics of attributes.

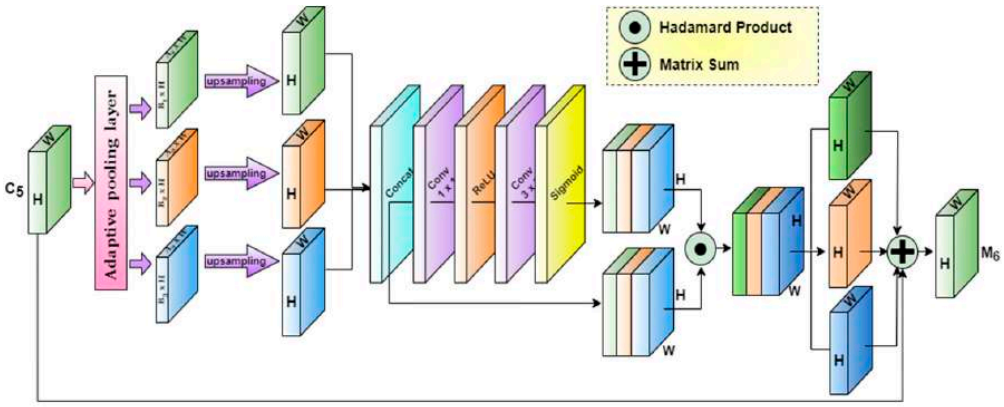



**Figure 7.** Proposed attention module with adaptive pooling.

In next stage, we focus on weighted feature generation which is obtained by applying multiple convolutions. The attention mechanism focuses on the informative part of the image. It is applied on image to obtain features or a single cross-sectional slice of tensor. With the help of these weight maps, the obtained context attributes are fused, and a new feature map is generated. According to this process, the newly generated feature maps are generated and combined with the high-level feature map and this set of feature map is propagated to the lower-level features.

According to the architecture of attention module, the input size is of $C_5$ is $H \times W$. This C5 is given as the input to the augmentation module. The context attributes are given as input to this module. These context features are obtained at different scaling levels such as $\beta_1 \times S$, $\beta_2 \times S$, and $\beta_3 \times S$ with the help of adaptive pooling. Adaptive pooling attempts to improve classic pooling approaches by introducing learned parameters within the pooling layer [29]. In [30], the authors reported the advantages of adaptive pooling for image segmentation tasks because several traditional algorithms use spatial pyramid pooling of local features which reduces the geometric information between regions. Adaptive pooling

is a technique which is used for multiscale summarization over generated convolution feature maps. This reduces the convergence time and increase the learning performance.

Therefore, this work uses $1 \times 1$ convolution to obtain the channel dimension of 256 followed by a bilinear interpolation to upsample the features up to S scale. Later, the attention mechanism is used to merge the context features by applying Concat layer and further these features are passed through the $1 \times 1$ convolution layer, ReLU activation layer, $3 \times 3$ conv layer and sigmoid based activation layer to obtain the weighted maps for each features map. Later, Hadamard product operation is applied on the obtained weighted features and feature maps which are further added to the $M_5$ to combine the context attributes with $M_6$. The features generated through this process contain rich and high quality attributes which helps to ensure the minimization of the information loss.

Similarly, the attribute augmentation process is also incorporated to improve the learning process. This approach uses dilation convolution to achieve the attributes adaptively for varied apple sizes by using adaptive pooling. This process is divided into two parts as multi-branch pooling layer where the average pooling layer is applied to fuse the image information received from three receptive fields and multi-branch convolution layers which are used to generate the feature maps with varied size of receptive field by using dilated convolution. This mechanism helps to improve the overall detection accuracy for small apples in the given image. The given Figure 8 shows the proposed arrangement of the attribute augmentation model.

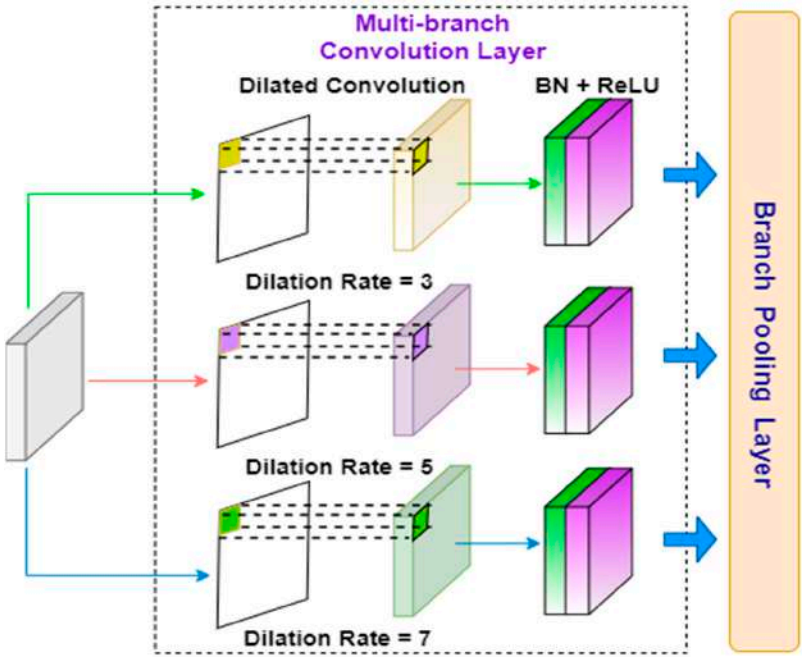



**Figure 8.** Attribute augmentation model.

According to Yang et al. [31], the shallow attributes are progressively enhanced with the help of semantics of deep features. This shows the importance of top-level attributes. Similarly, in [32], the authors also adopted a context feature enhancement model to obtain the scale attribute information to improve the small object detection. This model uses dilated convolutions which helps to process the different scales of the object. This model can be divided into two sub-models, a multi-branch convolution layer and a pooling layer. Here, multi-branch convolution plays an important role because it provides different sizes of receptive fields for the given multiscale high quality feature map by using dilation convolution. Finally, an average pooling layer is applied to fuse these attributes. This layer contains a dilated convolution, batch normalization and ReLU activation layer. The dilation convolution has three parallel branches with a kernel size of $3 \times 3$ and the dilation rate is

set as 3, 5, and 7 for each parallel branch, respectively. The average branch fuses feature during training which makes effective testing module by passing the attributes through single branch. This averaging operation is expressed as:

$$y_p = \frac{1}{B}\sum_{i=1}^{B} y_i \tag{1}$$

where $y_p$ represents the output of pooling layer branch and $B$ denotes the total number of considered parallel branch.

### *2.4. Box Prediction and Loss Function*

The bounding box prediction is the task of the neck module in the proposed YOLO architecture. The ground truth of the bounding box is denoted as $G = (x_1, y_1, x_2, y_2)$. Based on these points, the boundaries of G are computed as follows [33]:

$$t_{x1} = \log\frac{(s_l(x+0.5)-x_1)}{r_l},\ t_{y1} = \log\frac{(s_l(y+0.5)-y_1)}{r_l},\ t_{x2} = \log\frac{x_2 - s_l(x+0.5)}{r_l},\ t_{y2} = \log\frac{y_2 - s_l(y+0.5)}{r_l} \tag{2}$$

where $s_l$ denotes the scaling factor, $r_l$ is the basic scale, the coordinates $(x, y)$ are mapped to original image by applying down sampling later, projection coordinates and ground truth boxes are considered to estimate the normalized offset between the coordinates. At this stage, regularization is incorporated by applying the log-space function. Later, the smooth L1 loss function is used to train the loss function, similarly, $L_{reg}$ is applied for bounding box prediction. In next step, we focus on the loss function. Generally, the loss function helps to improve the target detect accuracy via an iterative optimization [34]. The target loss detector loss function contains two main components as classification and regression. The classification loss $L_{cls}$ is between confidence and regression loss is between regularized border and regression targets. The loss function is expressed as follows:

$$L(\{p_{si}\},\{t_i\}) = L_{cls} + L_{reg}\frac{1}{N_{cls}}\sum_i L_{cls}(p_{si}, p_i) + \lambda\frac{1}{N_{reg}}\sum_i p_i L_{reg}(t_i, t_i) \tag{3}$$

where $p_{si} = \begin{cases} p_i \; if \; p_i = 1 \\ 1 - p_i \; otherwise \end{cases}$, $\alpha_s = \begin{cases} \alpha \; if \; p_i = 1 \\ 1 - \alpha \; otherwise \end{cases}$ and $C = \begin{cases} 1 \left| t_{ij} - t_{ij} \right| < 1 \\ 0 \; otherwise \end{cases}$.

Here, $\alpha$ is used to balance the positive and negative sample imbalance which is caused due to a smaller number of samples of target image, i.e., samples of apple images are fewer with respect to the entire image. Thus, the model achieves the accurate bounding boxes resulting in improved accuracy.

The proposed focal loss function helps to estimate the classification loss, then $\alpha$ is used to balance the impact of positive and negative loss function. Moreover, it also avoids the dominance of classification loss generated by the samples. In order to find the bounding boxes, L1 loss is adopted to estimate the regression loss and $\beta$ helps to select the L1 or L2 loss function according to the loss range. This is useful in avoiding the slow convergence of L1 loss and sensitivity of L2 loss to outliers. Further, these loss functions are regularized by the $N_{reg}$ and $N_{cls}$. Finally, the total loss $L$ is backpropagated in gradient manner to update the model parameters and the final optimal model is generated.

## 3. Results and Discussion

This section presents the outcome of proposed Yolov5 architecture. This model was tested on real-time images of apple orchards. The proposed approach was implemented in the Ubuntu 16.04 operating system with the help of PyTorch deep learning framework for apple detection. The operating system was installed with Intel i7 processor, 24GB RAM, NVIDIA GeForce RTX 3090 connected with 384-bit memory interface. The operation of the GPU was at a frequency of 1395 MHz. The complete model was written in python programming language. This model uses YOLOv5 and improved its architecture and

performance of each model with the help of CUDA toolkit and CUDNN library. The complete experiment is carried out for a IoU threshold of 0.75.

### *3.1. Performance Evaluation*

In order to evaluate the performance of proposed approach three indicators, namely precision, recall and F1-score, plus the accuracy. These parameters can be computed as follows:

$$Pr = \frac{T_P}{F_p + T_P}, \quad Rec = \frac{T_P}{T_P + F_N}, \quad F1 - score = \frac{2PrRec}{Pr + Rec} \tag{4}$$

where, $T_p$, $F_p$, and $F_N$ denotes the true positive, false positive, and false negative values.

### *3.2. Apple Detection Performance*

This section presents the outcome of proposed approach to detect the apples in the given image. The obtained performance is compared with the other models. In order to show the robustness of proposed model, we have considered several factors which affect the performance of system such as illumination variation, blurred images and noise. The blurriness stage uses Gaussian blur model with ($3 \times 3$) kernel. Similarly, for illumination variations we have used Python's PIL package where 0.5 factor is assigned to obtain low illumination images and 1.5 factor is used for bright images. This section presents the comparative analysis of the proposed approach in terms of precision, recall and F1-score for the original image. The obtained performance of the proposed approach is compared with existing schemes such as AlexNet, ResNet, Faster RCNN, AlexNet + Faster RCNN, ResNet + FasterRCNN, YOLOv3, Improved Yolov3, and YOLOv5. The given Table 1 shows the comparison of detection performance.

**Table 1.** Comparative performance for original images and illumination variations.

| | **Original Images** | | | **Illumination Variations** | | |
|---|---|---|---|---|---|---|
| **Detection Method** | **Precision** | **Recall** | **F1-Score** | **Precision** | **Recall** | **F1-Score** |
| AlexNet | 0.66 | 0.72 | 0.69 | 0.61 | 0.65 | 0.63 |
| ResNet | 0.72 | 0.76 | 0.74 | 0.72 | 0.69 | 0.7 |
| Faster RCNN | 0.83 | 0.79 | 0.81 | 0.68 | 0.71 | 0.7 |
| AlexNet + Faster RCNN | 0.88 | 0.84 | 0.86 | 0.68 | 0.75 | 0.71 |
| ResNet + FasterRCNN | 0.87 | 0.64 | 0.74 | 0.72 | 0.75 | 0.73 |
| YOLOv3 | 0.82 | 0.86 | 0.84 | 0.7 | 0.78 | 0.7378 |
| Improved Yolov3 | 0.83 | 0.9 | 0.86 | 0.81 | 0.86 | 0.83425 |
| YOLOv5 | 0.89 | 0.97 | 0.93 | 0.85 | 0.91 | 0.87897 |
| Improved Yolov5 | 0.97 | 0.99 | 0.98 | 0.86 | 0.93 | 0.89363 |

The given Figure 9 depicts the detection performance by using the proposed approach. Figure 9a,d shows the sample input images and Figure 9b,e depicts the detection by Yolov5 and Figure 9c,f shows red delicious apple detection by proposed Yolov5. As depicted in the images below the number of apples detected using improved YOLOv5 is greater than the apples detected by Yolov5.

The Figure 10a,d shows the sample input images and Figure 10b,e depicts the detection by Yolov5 and Figure 10c,f shows gold delicious apple detection by proposed Yolov5. As depicted in the images below, the number of gold delicious apples detected using YOLOv5 is less than that of the improved YOLOv5.

(**a**)

(**b**)

(**c**)

**Figure 9.** *Cont*.

(**d**)

(**e**)

(**f**)

**Figure 9.** Apple (red delicious) detection outcome for normal images; (**a**). Apple (red delicious) original image 1 (frame 750); (**b**) apple (red delicious) detection by original Yolov5; (**c**) apple (red delicious) detection by proposed Yolov5; (**d**) apple (red delicious) original image 2 (frame 750); (**e**) apple (red delicious) detection by original Yolov5; (**f**) apple (red delicious) detection by proposed Yolov5.

(**a**)

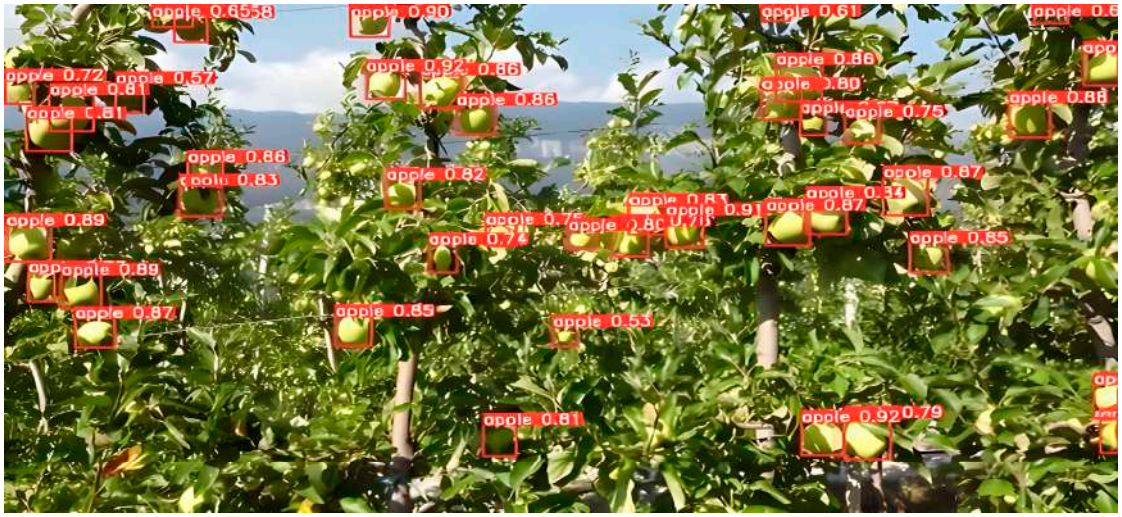
(**b**)

(**c**)

**Figure 10.** *Cont.*

(**d**)

(**e**)



(**f**)

**Figure 10.** Apple (gold delicious) detection outcome for normal images; (**a**) apple (gold delicious) original image 1 (frame 11640); (**b**) apple (gold delicious) detection by original Yolov5; (**c**) apple (gold delicious) detection by proposed Yolov5; (**d**) apple (gold delicious) original image 2 (frame 12237); (**e**) apple (gold delicious) detection by original Yolov5; (**f**) apple (gold delicious) detection by proposed Yolov5.

## 4. Discussion

Several of the methods used so far test the novelty and accuracy and the deep learning scheme, Adaptive Training Sample Selection (ATSS) for apple fruit detection. This approach to terrestrial RGB imagery functions at close range and at low cost. The main advantage of this method is that it only labels the center of the apple and not the bounding box. This is a great advantage in dense orchards. In addition to the state-of-the-art technology, Yolov5 is reported to achieve better accuracy than other models. However, the performance of these systems is affected by background complexity, motion blur, low light, etc. To overcome these problems, we introduce a new deep learning approach based on Yolov5. On top of this architecture, we integrated an attribute expansion model and an adaptive pooling operation to handle size variations. Validation of the system was tested on live agricultural field to confirm the apple detection inferences. The resulting output is reviewed by the end user and some ground truth issues in sunlight and shade are observed. These issues were later resolved by changing the confidence threshold and non-maximum suppression threshold, which returned the best results.

Further, we have considered the illumination variation and noisy images as inputs and measured the performance in terms of precision, recall and F1-score. The given Table 2 shows the performance for blurred and noisy image data.

**Table 2.** Comparative performance for Blurred Images and Noisy Images.

| | **Blurred Images** | | | **Noisy Images** | | |
|---|---|---|---|---|---|---|
| **Detection Method** | **Precision** | **Recall** | **F1-Score** | **Precision** | **Recall** | **F1-Score** |
| AlexNet | 0.69 | 0.87 | 0.77 | 0.65 | 0.71 | 0.75 |
| ResNet | 0.72 | 0.85 | 0.77 | 0.76 | 0.73 | 0.75 |
| Faster RCNN | 0.76 | 0.86 | 0.8 | 0.79 | 0.72 | 0.76 |
| AlexNet + Faster RCNN | 0.8 | 0.86 | 0.83 | 0.78 | 0.76 | 0.77 |
| ResNet + FasterRCNN | 0.8 | 0.88 | 0.84 | 0.78 | 0.83 | 0.8 |
| YOLOv3 | 0.84 | 0.86 | 0.85 | 0.8 | 0.87 | 0.83 |
| Improved Yolov3 | 0.88 | 0.9 | 0.89 | 0.82 | 0.9 | 0.86 |
| YOLOv5 | 0.94 | 0.83 | 0.88 | 0.87 | 0.97 | 0.92 |
| Improved Yolov5 | 0.96 | 0.99 | 0.99 | 0.92 | 0.97 | 0.94 |

The comparative studies show that the proposed approach achieves significant performance even in blurred, noisy and low illumination scenes. The corresponding time complexity with each of different models of Yolov5 with the accuracy of mAP 0.5, CPU time and GPU time is depicted in Table 3.

**Table 3.** Comparative analysis performance of time complexity on Yolov5 models.

| **YoloV5 Models** | **Parameters (Million)** | **Accuracy (mAP 0.5)** | **CPU Time (ms)** | **GPU Time (ms)** |
|---|---|---|---|---|
| YOLOv5x | 42.1 | 69.72 | 710 | 15.6 |
| YOLOv5I | 31.5 | 76.30 | 330 | 13.3 |
| YOLOv5m | 26.2 | 81.20 | 240 | 9.2 |
| YOLOv5s | 18.3 | 86.51 | 160 | 8.1 |
| Proposed YOLO Model | 11.1 | 91.25 | 72 | 6.1 |

As per the table with experimented results, the proposed Yolov5 model gives the optimum results with best accuracy, low CPU and GPU time.

Further, we present a k-fold validation study to measure the performance of proposed approach to check the robustness of proposed approach. The given Table 4 demonstrates the outcome for 5 kFold cross validations.

**Table 4.** Performance analysis of proposed approach for 5 kFold experiments.

| | **Original Images** | | | **Noisy Images** | | |
|---|---|---|---|---|---|---|
| **Folds** | **Precision** | **Recall** | **F1-Score** | **Precision** | **Recall** | **F1-Score** |
| kFold 1 | 0.96 | 0.97 | 0.99 | 0.95 | 0.97 | 0.94 |
| kFold 2 | 0.95 | 0.98 | 0.97 | 0.96 | 0.96 | 0.95 |
| kFold 3 | 0.96 | 0.98 | 0.96 | 0.95 | 0.96 | 0.96 |
| kFold 4 | 0.96 | 0.97 | 0.98 | 0.95 | 0.98 | 0.96 |
| kFold 5 | 0.98 | 0.96 | 0.97 | 0.97 | 0.97 | 0.98 |
| | Blurred Images | | | Illumination variation | | |
| Folds | precision | Recall | F1-score | precision | Recall | F1-score |
| kFold 1 | 0.95 | 0.96 | 0.98 | 0.98 | 0.96 | 0.97 |
| kFold 2 | 0.96 | 0.96 | 0.95 | 0.96 | 0.98 | 0.96 |
| kFold 3 | 0.97 | 0.99 | 0.97 | 0.98 | 0.97 | 0.94 |
| kFold 4 | 0.96 | 0.98 | 0.97 | 0.95 | 0.98 | 0.96 |
| kFold 5 | 0.95 | 0.96 | 0.94 | 0.96 | 0.97 | 0.93 |

According to this kFold analysis, we show that the proposed approach achieves good performance by achieving average performance as 0.96, 0.971, 0.968, 0.961, 0.97, and 0.955 in terms of precision, recall, F1-score, precision, recall, and F1-score, respectively.

## 5. Conclusions

In this work, we introduce a deep learning-based approach of apple detection and localization. We have focused on apple orchards, where the detection performance is highly important. In this field, accurate detection, localization and tracking are important tasks. A deep learning-based model has been introduced which is based on the Yolov5 architecture. The features generated through the attention mechanism contain rich and high-quality attributes which helps to ensure the minimization of loss of information. Similarly, we incorporated an attribute augmenting mechanism to achieve better accuracy for smaller objects. The comparative study shows that the proposed approach achieves overall performance with optimized results. In future, this model can be tested for apple quality check purposes by training the architecture for a different dataset.

**Author Contributions:** Conceptualization, P.K.S., F.M. and J.M.; methodology, P.K.S. and F.M.; software, J.M. and P.K.S.; formal analysis, P.K.S., F.M. and J.M.; resources, F.M. and J.M.; data curation, P.K.S. and F.M.; writing—original draft preparation, P.K.S. and F.M.; writing—review and editing, P.K.S. and F.M.; and supervision, F.M. and J.M.; project administration, J.M.; and 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'.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**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).

**Conflicts of Interest:** The authors declare no conflict 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.

## Abbreviations

| | |
|---|---|
| DL | Deep Learning |
| RGB | Red, Green and Blue |
| HSV | Hue, Saturation and Value |
| RGB-D | Depth Camera |
| R-CNN | Regions with Convolutional Neural Network |
| SIFT | Scale Invariant Feature Transform |
| RPN | Region proposal networks |
| HRNet | High-Resolution Network |
| RNN | Recurrent Neural Networks |
| CNN | Convolutional Neural Network |
| RPA | Remote Piloted Aircraft |
| SVM | Support Vector Machines |
| YOLO | You Only Look Once |
| IoU | Intersection over Union |
| ROI | Region of interest |
| ATSS | Adaptive Training Sample Selection |
| CMOS | Complementary Metal Oxide Semiconductor |
| RTMP | Routing Table Maintenance Protocol |
| CSP | Cross stage partial connections |
| Mp | Megapixel |
| µm | Micrometer |
| mAP | Mean average precision |
| VIA | VGG Image Annotator |
| VGG | Visual Geometry Group |
| ResNet | Residual Network |
| FPN | Feature Pyramid Network |
| P | Precision |
| R | Recall |
| std | Standard Deviation |
| BB | Bounding Box |
| BN | Batch Normalization |
| PANet | Path Aggregation Network |
| SGD | Stochastic Gradient Descent |
| AP | Average Precision |
| TP | True Positive |
| FP | False Positive |

## References


1. Zou, Z.; 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. Chandra, A.L.; Desai, S.V.; Guo, W.; Balasubramanian, V.N. Computer vision with deep learning for plant phenotyping in agriculture: A survey. *arXiv* **2020**, arXiv:2006.11391.
4. 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]
5. Kuznetsova, A.; Maleva, T.; Soloviev, V. Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot. *Agronomy* **2020**, *10*, 1016. [CrossRef]
6. Jia, W.; Zhang, Y.; Lian, J.; Zheng, Y.; Zhao, D.; Li, C. Apple harvesting robot under information technology: A review. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420925310. [CrossRef]
7. Jiao, Y.; Luo, R.; Li, Q.; Deng, X.; Yin, X.; Ruan, C.; Jia, W. Detection and localization of overlapped fruits application in an apple harvesting robot. *Electronics* **2020**, *9*, 1023. [CrossRef]
8. Li, T.; Fang, W.; Zhao, G.; Gao, F.; Wu, Z.; Li, R.; Dhupia, J. An improved binocular localization method for apple based on fruit detection using deep learning. *Inf. Process. Agric.* **2021**, *in press*. [CrossRef]
9. Gené-Mola, J.; Vilaplana, V.; Rosell-Polo, J.R.; Morros, J.R.; Ruiz-Hidalgo, J.; Gregorio, E. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. *Comput. Electron. Agric.* **2019**, *162*, 689–698. [CrossRef]
10. Gené-Mola, J.; Gregorio, E.; Guevara, J.; Auat, F.; Sanz-Cortiella, R.; Escolà, A.; Rosell-Polo, J.R. Fruit detection in an apple orchard using a mobile terrestrial laser scanner. *Biosyst. Eng.* **2019**, *187*, 171–184. [CrossRef]

11. Liu, Q.; Zhao, X.; Yang, H.; Zhao, L.; Ling, W.; Ma, X.; Zhao, Y. Image segmentation of Huaniu apple based on pulse coupled neural network and watershed algorithm. In Proceedings of the International Conference on Electronic Information Engineering and Computer Communication (EIECC 2021), Nanchang, China, 17–19 December 2021; Volume 12172, pp. 448–455.
12. Zhang, C.; Zou, K.; Pan, Y. A method of apple image segmentation based on color-texture fusion feature and machine learning. *Agronomy* **2020**, *10*, 972. [CrossRef]
13. Yang, M.; Kumar, P.; Bhola, J.; Shabaz, M. Development of image recognition software based on artificial intelligence algorithm for the efficient sorting of apple fruit. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 322–330. [CrossRef]
14. Chen, S.W.; Shivakumar, S.S.; Dcunha, S.; Das, J.; Okon, E.; Qu, C.; Kumar, V. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robot. Autom. Lett.* **2017**, *2*, 781–788. [CrossRef]
15. Dias, P.A.; Tabb, A.; Medeiros, H. Apple flower detection using deep convolutional networks. *Comput. Ind.* **2018**, *99*, 17–28. [CrossRef]
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
17. Farhadi, A.; Redmon, J. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.0276.
18. Biffi, L.J.; Mitishita, E.; Liesenberg, V.; Santos AA, D.; Gonçalves, D.N.; Estrabis, N.V.; Gonçalves, W.N. ATSS deep learning-based approach to detect apple fruits. *Remote Sens.* **2020**, *13*, 54. [CrossRef]
19. www.personaldrones.it. Available online: https://www.personaldrones.it/341-mavic-3 (accessed on 1 September 2021).
20. www.dji.com. Available online: https://www.dji.com/it/mavic-3 (accessed on 1 September 2021).
21. 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.
22. 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.
23. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef]
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
25. Ghaffarian, S.; Valente, J.; Van Der Voort, M.; Tekinerdogan, B. Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review. *Remote Sens.* **2022**, *13*, 2965. [CrossRef]
26. Hu, J.; Zheng, Y.; Lam, K.M.; Lou, P. DWANet: Focus on Foreground Features for More Accurate Location. *IEEE Access* **2022**, *10*, 30716–30729. [CrossRef]
27. Wang, S. Research Towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications. *J. Phys. Conf. Ser.* **2021**, *1948*, 012021. [CrossRef]
28. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
29. Wolter, M.; Garcke, J. Adaptive wavelet pooling for convolutional neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 13–15 April 2021; pp. 1936–1944.
30. Tsai, Y.H.; Hamsici, O.C.; Yang, M.H. Adaptive region pooling for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 731–739.
31. Yang, X.; Liu, Q. Scale-sensitive feature reassembly network for pedestrian detection. *Sensors* **2021**, *21*, 4189. [CrossRef] [PubMed]
32. Zhu, X.; Liang, B.; Fu, D.; Huang, G.; Yang, F.; Li, W. Airport small object detection based on feature enhancement. *IET Image Process.* **2021**, *16*, 2863–2874. [CrossRef]
33. 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]
34. 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] [PubMed]