Multi-Class Flower Counting Model with Zha-KNN Labelled Images Using Ma-Yolov9

Multi-Class Flower Counting Model

A. Jasmine Xavier[1], S. Valarmathy[2], J. Gowrishankar[3], B. Niranjana Devi[4]
Department of Electronics and Communication Engineering, Jayaraj Annapackiam CSI College of Engineering, Nazareth, India[1]
Department of Electronics and Communication Engineering, Vinayaka Mission's Kirupananda
Variyar Engineering College, Salem, India[2]
Department of Computer Science and Engineering (Artificial Intelligence), JAIN (Deemed-To-Be University), Bangalore, India[3]
Department of Biomedical Engineering, Paavai Engineering College, Pachal, Namakkal, India[4]

***Abstract*—The flowering period is a critical time for the growth of plants. Counting flowers can help farmers predict the corresponding fields' yield information. As there are several works proposed for flower counting purposes, they lack the prediction of different flowers with counts. Hence, a novel model has been proposed in this study. Initially, this model is fed with different flower images as input, then these images undergo pre-processing. In pre-processing, the images are converted to grayscale for improved accuracy, and then the image's noise is removed using bilateral filters. Noise-removed images are then given for edge detection, using GI-CED. Edge-detected images are then augmented to improve the learning rate of the model. Augmented images are labeled using ZHA-KNN. Labeled images feature extracted and are given to MA-YoloV9, which is pre-trained to detect flowers in the image count and obtained as output. Overall, the proposed model was implemented and obtained an accuracy value of about 98.8% and F1-Score obtained 92.2% which is far better than the previous counting models.**

Keywords*—Flower counting; bilateral filter; Zhang Shasha Algorithm distance measured-K-Nearest Neighbor (ZSA-KNN); Gradient Intensity-Canny Edge Detection (GI-CED); mish-activated YoloV9

I. INTRODUCTION

Flower counting is used for estimating the yield and selecting favorable genotypes of particular crops [1]. It helps farmers and producers to allocate the necessary resources during the harvesting season [2]. Flower count fluctuates significantly throughout cultivars, locations, and seasons [3]. Furthermore, the economic planning and market forecasting of farmers can benefit from early yield predictions based on flower counting [4, 5]. To quicken the counting process, in manual counting, flowers in clusters were counted rather than the individual flowers. It is assumed that there are many flowers in a single cluster, but this might not be true for all the flower clusters [6]. There is also a high probability that few flower clusters might be counted twice or not all counted [7]. Additionally, due to occlusion the flower clusters can become unobservable.

Flower fragmentations, flower merges, false detections, and missing detections are further consequences of flower detection errors. Such errors in detection result in inaccurate counts of flowers [8]. Conventional flower counting often needs regular observation and expert knowledge and entails manual estimating by researchers based on their skill. This method requires a lot of work and is prone to mistakes. To precisely measure the amount of flowers on different trees, researchers use a variety of image sensors and algorithms to tackle the flower counting challenge. Conventional image processing methods are usually used by using color thresholds to categorize the pixels in pictures of flowers taken from fruit trees. As a result, a correlation can be established between the fraction or quantity of pixels designated as target flowers and the actual number of flowers. Unfortunately, this method's accuracy and generalizability are limited because it is sensitive to ambient lighting and necessitates rigorous environmental control [9, 10].

DeepLab-ResNet, Mask R-CNN, and Fully Convolutional Neural Networks (FCNN) can segment flowers, however they are less good at recognizing big aggregations of flowers and are unable to count the number of flowers [11]. Thus, many research use Deep Learning (DL) to count flowers in agriculture [12, 13]. DL is employed more and more for image segmentation and classification because of its capacity to develop robust discriminants, handles the counting process. For preprocessing, they need pre-labelled datasets, which are utilized for image-level classification or flower identification [14]. DL techniques enable automated feature recognition in the absence of human intervention, as the networks are trained to autonomously identify the most significant features [15].

The foremost contribution of the analysis is:

- Previous works have not focused on counting different flower images. Hence a novel work has been proposed in this study that even can count different varieties of flowers.
- Noised images directly from the field created difficulty in the detection of flowers and also counting them other than leaves. For this purpose, a bilateral filter along with Gradient Intensity- Canny Edge Detection (GI-CED) has been proposed.
- As number of flower image dataset was low previously it was difficult to train and get accurate output. Hence data augmentation with labelling using Zhang Shasha

Algorithm Distance Measured- K-Nearest Neighbor (ZSA-KNN) has been proposed.

- Previous object detection models were not satisfactory in detecting robust counting hence Yolo V9 has been taken in this study also to improve its training Mish activation function has been modified.

The outline of this analysis is followed as Section II lists the related work in counting flowers, Section III covers the proposed work, and the findings are explained in Section IV. The conclusion is explained in Section V.

II. Literature Review

Flower counting is an interesting problem discussed in literature which is used for both crop yield output and boutique flower counting. We have discussed a few related works in this section.

Dragon fruit flowers, mature fruits, and immature fruits were classified and counted by video stream inspection type by Li et al., [16]. The procedure involved tri crucial steps. The correct identification of the flowers was proved by You Only Look Once (YOLOv5) detection model. Real-time counting capability reached 56 frames per second of counting frequency. However, Matthews correlation coefficient (MCC) was not examined

Unmanned Aerial Vehicle (UAV)-based Deep Learning (DL) counting technique was inspected by Li et al., [17]. The density estimation problem for the in-field counting of rape flower clusters was developed. The two datasets used for network model training were Rape Flower Rectangular Box Labelling (RFRB) and Rape Flower Center Point Labelling (RFCP). The suggested networks included the benefits of popular regression estimation techniques, which hang on strong feature extraction capabilities. Research on the meadow of rape flower was not held.

Lightweight model that compresses the YOLOv5 network through filter pruning was provided by Yu et al., [18]. An adaptive batch normalization layer evaluation mechanism was contained within pruning method to assess the subnetwork's presentation. YOLOv5_E network was effectively applied for agricultural equipment such as UAV for the detection of crops on handheld apparatuses. F1-score metrics was not reviewed.

Discovered that Faster Region-Based Convolutional Neural Network (R-CNN) model performed the best when a range of DL algorithms were examined for the identification and counting of soybean flowers and pods by Zhu et al., [19]. Faster R-CNN model underwent additional refinement and optimization as per the features of soybean flowers and pods. However, Different varieties of flowers were not considered.

Machine Learning (ML) techniques involved using the Rotation, Scaling and Translation (RST)-Invariant Feature, Pattern Classification, and K-Closest Neighbor computations was suggested by Kaur et al., [20]. K- Nearest Neighbor (KNN) was used to separate six dissimilar sets of sunflower pictures with effectiveness. Better accuracy was needed for accurate results. The detection algorithm needed to be improved for accurate results.

Create a strategy based on DL to describe cotton plant blossoming patterns by Jiang et al., [21]. Statistics and regression analyses demonstrated that imaging-derived flowering characteristics were just as effective as manual assessment in distinguishing genetic categories or genotype differences and that the Deep-Flower method was identified and counted emerging flowers on cotton plants. Low accuracy was identified.

Combined counting technique was suggested by Yang et al., [22] through several angles of view with a DL algorithm. A dynamic head framework and a two-scale recognition branch was created based on YOLOv5. The results indicated that the identified and counted flowers was even in complex occlusion. Overlapping flowers were difficult to detect.

Explored the application of YOLOv9 by Vo et al., [23] in the arrangement of tomato ripeness stages and numeration tomatoes. Using common assessment metrics like Precision, Recall, and mAP50 was assessed the suggested model's performance and provided important insights into how well it detected and counted tomatoes in hands-on situations. Different DL architectures were not incorporated.

The performance of an image processing method intended to count and recognize oranges was examined by Shankarpure et al., [24]. The algorithm's output showed that an average accuracy rate was attained. The outcome demonstrated that the algorithm performs effectively with fewer fruits. More performance metrics were not considered.

A novel image-processing framework and an enhanced YOLOv8 model was employed by Wang et al., [25]. The number of strawberry plants in a total of 100 images was determined by a newly accessible image-processing system. Images taken through whole development cycle allowed to compute the harvest index for various locations. Varieties of fruit flowers was not considered.

Dandelion flowers were counted by Patton et al., [26] in lawn images via freely available software. The particle analysis function was recognized to discriminate dandelion flowers inside the open-source software ImageJ. 164 images were inspected to detect the accuracy. However, Precision, recall was not considered. YOLOv5 was presented by Viveros Escamilla et al., [27] for the detection and classification of full growth of bell peppers. D435i RGB-D camera was integrated also the tracking method was accessed by Multiple-Object Tracking (MOTA). The Deep Simple Online and Realtime (DeepSORT) algorithm was applied for sweet peppers tracking. Achieved accurate outcomes still, robustness was decreased.

Faster Region-based Convolutional Neural Network (Faster R-CNN) was finetuned, subsequently, a color-based thresholding process was presented by Rahim et al., [28] to count tomato flowers in greenhouses. Investigation performed with different categories of datasets to detect accurate flower of different flower measures. Still, Small flowers were unable to detect. YOLO-deepsort was suggested by Ge et al., [29] to identify and count tomatoes in dissimilar growth periods. Tomato flowers was counted by suggested method in complicated agricultural cases. Cosine annealing algorithms were used to improve the learning rate. However, the counting

and tracking were erroneous because of shaking in the scrutiny process.

Phenology distribution estimation method titled as Deep Phenology for apple flowers based on CNNs by RGB images was suggested by Wang et al., [30]. Visual Geometry Group 16 (VGG- 16) was immediately trained with relative phenology distributions considered from labor counts of flowers in the orchard. The suggested approach avoided the confrontation of differentiation of overlapped flower clusters. Precision metrics were not considered.

III. PROPOSED METHODOLOGY

Previously there were several flower counting works proposed, still, there are few works that did not detected the type of flower and their counting effectively. The proposed model performs both classifications of type of flower and results in their count the block diagram of the proposed model is shown in Fig. 1. The canny edge detection method is selected, which is modified using gradient intensity to select its maximum and minimum. Hence it is termed Gradient Intensity-Canny Edge Detection (GI-CED). Due to augmentation, clustering of images is necessary, as this helped to increase the training accuracy. This cluster labeling is used to annotate images thereby flower images get labeled as per their names. For this purpose, the Zhang Shasha Algorithm Distance Measured- K-Nearest Neighbor (ZSA-KNN).

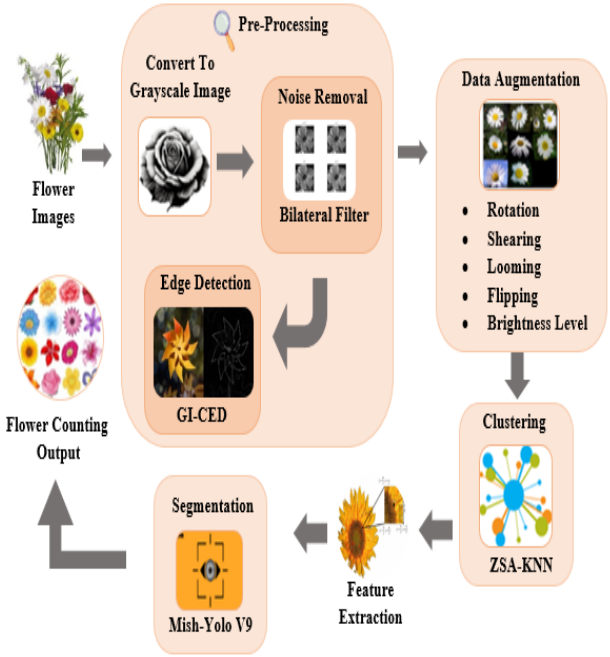

Fig. 1. Block diagram of the proposed model.

A. Data Collection

Initially, flower images of flowers, namely, 'Common Lanthana', 'Hibiscus', 'Jatropha', 'Marigold', 'Rose', 'champaka', 'chitrak', 'honeysuckle', 'Indian mallow', 'Malabar melastome', 'shanku pushpam', 'spider lily' and 'sunflower' has been taken. This can be given as:

$$D_C = \{D_1, D_2, ...D_T\} \quad (1)$$

In this equation (1), D_C denotes the total number of flower data images in the set.

B. Pre-Processing

Pre-processing is an important step in image processing. The steps involved in this pre-processing are greyscale conversion, noise removal, and edge detection.

1) Conversion to greyscale: Greyscale conversion is important in object categorization as it can enhance visualization. Also, it helps to differentiate between highlights and shadow details.

2) Noise removal: Noise in the image is removed using a bilateral filter. Bilateral filters can blur an image while preserving strong edges. Processing noisy images using a bilateral filter is given as:

$$B_{im} = \sum_{u=a-R}^{a+R} \sum_{v=b-R}^{b+R} C(a,b;u,v) D_C^{u,v} \quad (2)$$

Where B_{im} denotes the processed pixel at (a,b), $C(a,b;u,v)$ is the weight coefficient between a current pixel and its neighboring points.

3) Edge detection: Image processing utilizes edge detection to pinpoint areas in a digital image in sudden shifts in brightness or intensity, known as edges or boundaries.

- Noise reduction was done already using a bilateral filter
- Gradient intensity is calculated using

$$I_{Gr} = \tan^{-1}\left(\sqrt{(\gamma_a)^2 + (\gamma_b)^2}\right) \quad (3)$$

In Eq. (3), γ_a represents the change in the image's x-axis and the image's y-axis of noise noise-removed image (B_{im}).

- Further, non-maximum suppression to reduce duplicate merging pixels along edges was done to make them uneven.
- Edge tracking was done as weak edge pixels caused by true edges will be connected to a strong edge pixel. All these connected components will be gradient mapped considering each pixel only once. Edge-detected images are represented as:

$$\mathrm{Im}_E = GI - CED\{I_{Gr}\} \quad (4)$$

In this Eq. (4), Im_E represents the edge detected image.

C. Image Augmentation

Image augmentation increases the diversity of the dataset and the robustness of the model. In this work augmentation performed are ‘rotation’, ‘shearing’, ‘zooming’, and ‘brightness level’. The image augmentation process can be given as

$$Aug \leftarrow \{'rotation','shearing','zoo\ min\ g','brightnesslevel'\} \quad (5)$$

$$\mathrm{Im}_{Aug}^{1} = Aug\{\mathrm{Im}_{E}\} \quad (6)$$

Equations (5) and (6), show the method of augmentation used in this work. The overall augmentation output can be given as:

$$\mathrm{Im}_{Aug}^{J} = \left(\mathrm{Im}_{Aug}^{1}, \mathrm{Im}_{Aug}^{2}, ...\mathrm{Im}_{Aug}^{N}\right) \quad (7)$$

Where, Im_{Aug}^{J} represents the overall argumentation output.

D. Clustering Using ZSA-KNN

Due to augmentation, clustering of images is necessary, as this helped to increase the training accuracy. This cluster labeling is used to annotate images thereby flower images get labeled as per their names. For this purpose, the Zhang Shasha Algorithm Distance Measured- K-Nearest Neighbor (ZSA-KNN). The steps of working ZSA-KNN are defined as follows.

Step 1**:** The input taken here $\mathrm{Im}_{Aug}^{J} = \left(\mathrm{Im}_{Aug}^{1}, \mathrm{Im}_{Aug}^{2}, ...\mathrm{Im}_{Aug}^{N}\right)$ and the set of clusters presented in the dataset is $L_{e=[1,2,..I]}$, the ZSA-KNN query of the point represented as

$$L_e(l) = \{l_f\},\ f = [1,2,3,...F] \quad (8)$$

Where, equation (8), represents $L_e(l)$ is the neighborhood of the points in the x-axis.

Step 2: If the data volume is defined by query the density of the cluster Le at the point in the x-axis is defined by the ratio of the number of ZSA-KNN. This can be given as:

$$F = \frac{size\left(Le(l) \cap \mathrm{Im}_{Aug}^{J}\right)}{V\left(\mathrm{Im}_{Aug}^{J}\right)} \quad (9)$$

Where, in Eq. (9), $V\left(\mathrm{Im}_{Aug}^{J}\right)$ denotes the volume of the cluster. Then the equation for labelling is given as

$$F\left(\mathrm{Im}_{Aug}^{J}\right) = \underset{f=[1,2,...F]}{Max}\left(f\left(\mathrm{Im}_{Aug}^{J}, Le\right)\right) \quad (10)$$

Based on this equation, the clustering is started. If there is more than one clusters containing the same max points, then the point has to be assigned to the closest cluster which is defined by the nearest distance defined using Zhang Shasha Algorithm Distance.

Step 3: Zhang Shasha Algorithm Distance is given as:

$$Dis\tan c\, e(l(c), l(d)) = min\{zhangshasha(l(c1)..c - 1..l(d1) + \vartheta[T[c]]), zhangshasha(l(c2)..c - 1..l(d2) + \vartheta[T[c]])... + \vartheta(T(c \to d)) \quad (11)$$

In this equation (11), the distances correspond to the cost of deleting and inserting cluster members denoted as $l(c1) and\ l(d1)$.

Step 4: Estimation of this distribution helps to find the perfect cluster. The data object to the right group of the dataset. At each iterative step, every point x is assigned to the nearest obtained cluster Le. Each point has a chance to be assigned or to be re-assigned to the most suitable cluster at this procedure. This step is repeated until convergence when no or only a few points are changing from one cluster to another. The labeled output can be given as

$$\mathrm{A}_l = \left|\mathrm{A}_l^1, \mathrm{A}_l^2, ...\mathrm{A}_l^m\right| \quad (12)$$

Where, Eq. (12), A_l is the total labels obtained.

Algorithm 1: Pseudo code for ZSA-KNN

Input: Augmented images and cluster numbers
Output: Labelled output

Begin
For $e \leftarrow 1tol$ do visit $[e] \leftarrow false$
Initialize the label as Le
$L_e(l) = \{l_f\},\ f = [1,2,3,...F]$
Do
$F\left(\mathrm{Im}_{Aug}^{J}\right) = \underset{f=[1,2,...F]}{Max}\left(f\left(\mathrm{Im}_{Aug}^{J}, Le\right)\right)$
If $(u = \max)$
Visited $[J] \leftarrow true$
Current $\leftarrow J$
$F = \frac{size\left(Le(l) \cap \mathrm{Im}_{Aug}^{J}\right)}{V\left(\mathrm{Im}_{Aug}^{J}\right)}$
Else
For $e \leftarrow 2tol$
Find lowest element using distance equation
$Dis\tan c\, e(l(c), l(d)) = min\{zhangshasha(l(c1)..c - 1..l(d1) + \vartheta[T[c]]), zhangshasha(l(c2)..c - 1..l(d2) + \vartheta[T[c]])... + \vartheta(T(c \to d))$
Current $\leftarrow J$
End for
Update cluster
Repeat until convergence
End if
Return cluster label
End for
End

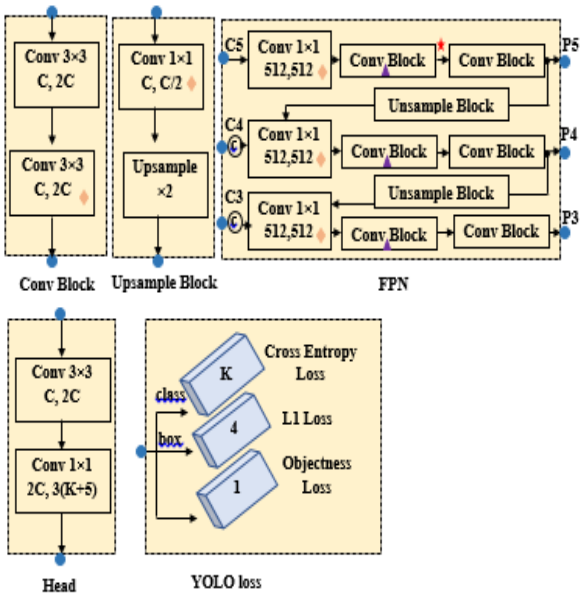

Fig. 2. Architecture of MA-YoloV9.

E. Feature Extraction

After clustering, the images are given for feature extraction. Features such as Convex Hull, Local Binary Patterns, Scale-Invariant Feature Transform, Hu, and Aspect Ratio are extracted. This can be given as

$$f_x = Fea(\mathrm{A}_l) \tag{13}$$

In Eq. (13), the features are extracted and represented as f_x .

F. Object Detection Using Mish-Activated Yolov9

Extracted features are then given for object detection using mish-activated YoloV9. YoloV9 is selected here for object detection due to its strong competitiveness, still, its performance in small objects was not satisfactory. Hence it is modified to Mish activation. The architecture of Mish-Activated-YoloV9 is shown in Fig. 2.

MA-YoloV9 used generalized ELAN which was a combination of ELAN and CSPNet. Features are given to the model along with the original images.

Hyper-interfering the images into overlapping patches that result in larger pixel areas for small objects, then extracting patches and resizing them into larger images. Each image is sliced into overlapping patches:

$$Z_g = \lfloor Z_{g1,g2,\ldots gt} \rfloor \tag{14}$$

Where, g is the total number of overlapping patches with dimensions $s \times t$ .Then the patches are resized taking care of aspect ratio. Finally, the overlapping prediction results are merged into the original image.

- This model has Programmable Gradient Information (PGI) which includes a main branch, auxiliary reversible branch, and multi-level auxiliary branch.
- Auxiliary reversible branch: In this branch it mitigates the inherent information loss within deep network layers, preserving and harnessing complete data for learning. The information loss is given as

$$\mathrm{N}(\Re,\Re) \geq \mathrm{N}(\Re, f_\theta(\Re)) \geq \mathrm{N}(\Re, g_\theta(f_\theta(\mathrm{N}))) \tag{15}$$

- In Eq. (15), N indicates mutual information, f *and* g are transformation functions, θ is the parameters of f *and* g respectively. When function ρ has an inverse transformation function, the reversible function is given as:

$$\mathrm{N} = \upsilon_\varsigma(\rho_\varphi(\mathrm{N})) \tag{16}$$

In Eq. (16), φ *and* ς are parameters of υ *and* ρ , respectively. The reversible function without losing information is given as:

$$\mathrm{N}(\Re,\Re) = I(\Re, \rho_\varphi(\mathrm{N})) = I(\Re, \upsilon_\varsigma(\rho_\varphi(\mathrm{N}))) \tag{17}$$

- Multi-level auxiliary information: For object detection, different feature pyramids can be used to perform different tasks, for example together they can detect objects of different sizes. Therefore, after connecting to the deep supervision branch, the shallow features will be guided to learn the features required for small object detection, and at this time the system will regard the positions of objects of other sizes as the background.
- Generalized Elan: In this phase, the combination of CSPNet and ELAN was done with gradient path planning. In this phase, the Mish activation function is taken. This can be given as:

$$\mathrm{N}\tanh(\ln(1+e^{\mathrm{N}})) \tag{18}$$

This architecture predicts the types of flowers in the images and their counts using this deep network. The output is given as M

Algorithm 2: Pseudo code for MA-YoloV9

Input: Features, Original Image
Output: Object detected output

Begin
For $g \leftarrow 1tot$
do
Compute reversible function
For $\mathrm{H} \leftarrow 1to\,num$
Compute transition
Accumulate filter outputs
If (accumulation< threshold)
Perform ELAN
Else
Output sub-window face
End If
End For
End for
End

IV. Results and Discussion

The proposed experiments were implemented using the deep learning framework in Python version. The simulation results indicate that the model achieved high accuracy in counting flowers across different species. The results demonstrate that the model effectively generalizes to various flower types, and exhibits its validity and potential for practical applications in floriculture and ecological monitoring.

A. Dataset Description

Annot Image Dataset: It is a collection of images, capturing various flowers and plants from multiple viewpoints. This dataset was specifically created to train and evaluate deep learning models for flower counting and detection and focus on agricultural applications. The dataset comprised a diverse selection of flowers, including Common Lantana, Hibiscus, Jatropha, Marigold, Rose, Champaka, Chitrak, Honeysuckle, Indian Mallow, Malabar Melastome, Shankupushpam, Spider Lily, and Sunflower. Each image in the dataset is accompanied by detailed annotations, providing accurate counts for each flower type present. These annotations are crucial for training models to recognize and count flowers accurately, facilitating precise performance evaluation in real-world agriculture.

Dataset link: cannot Dataset > Overview (roboflow.com)

B. Performance Analysis of the Proposed Model

Table I indicates the output of an object detection model. Each row corresponds to a specific class of object that the model has detected. The number in each row indicates how many instances of that particular class were detected in the image. Object detection precisely outlining and pinpointing where each object is located.

Fig. 3(a) indicates the input image given to the proposed object detection model MA-YoloV9 along with extracted features. Fig. 3(b) indicates the output of flower detection, with various flowers being identified and confidence scores provided for each identification. It recognizes different types of flowers and assigns a likelihood to its predictions.

TABLE I. Object Detection Output

Class name	Count
class Champaka	25
class Chitral	23
class Common Lantana	24
class Hibiscus	21
class honeysuckle	23
class Indian-mallow	31
class Jatropha	43
class Malabar Melastome	24
class Marigold	21
class Rose	21
class Shankupushpam	22
class Spider lily	9
class Sunflower	34

(a)

(b)

Fig. 3. (a) Input image, (b) Flower detection output.

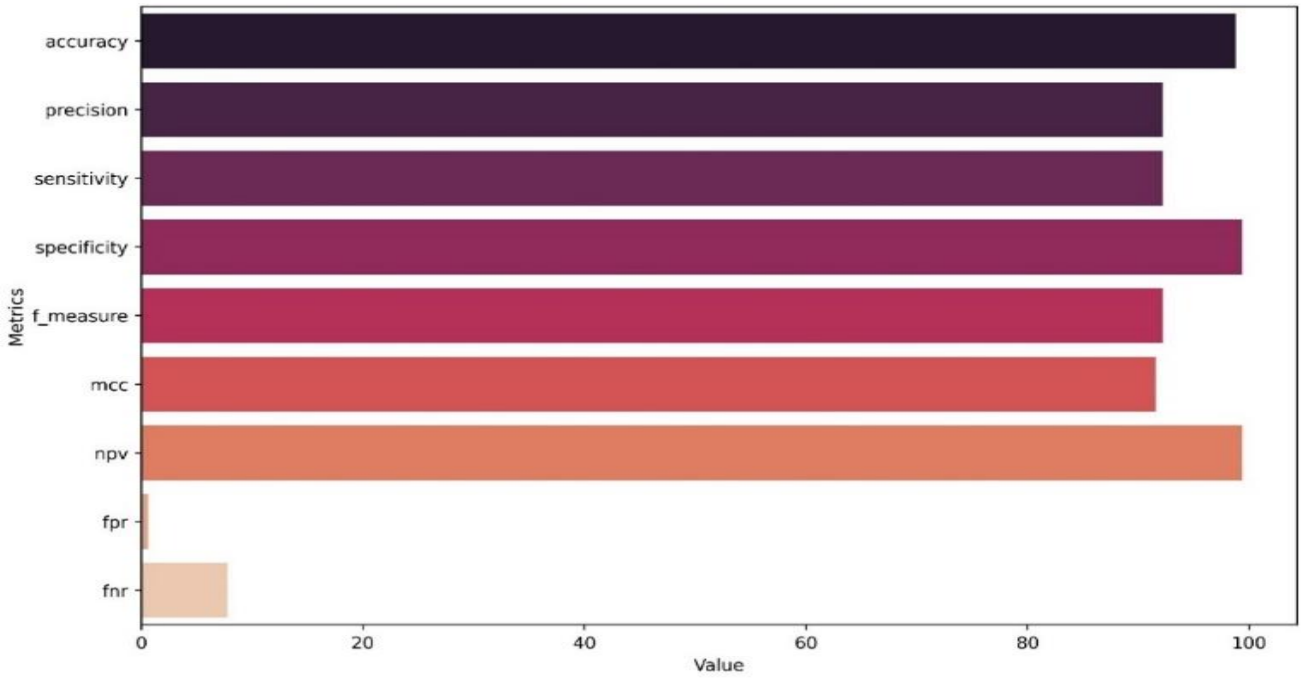

Fig. 4. Object detection model performance metrics.

Fig. 4 indicates the metrics collectively demonstrate the model's effectiveness. With an accuracy of 98.80%, the model demonstrates high overall correctness in distinguishing between different classes. Precision, with 92.22%, indicates a low false positive rate. Sensitivity, also known as recall, shares the same value as precision, highlighting the model's ability to effectively identify true positives. Specificity, at 99.35%, signifies the model's proficiency in correctly identifying true negatives. The F-measure with 92.21% further reinforces the model's balance between accurate positive predictions and comprehensive coverage of positive instances. The Matthews correlation coefficient (MCC), a metric that considers both false positives and false negatives, is impressively high at 91.57%, indicating strong agreement between predicted and actual flower counting output. The Negative Predictive Value (NPV) stands at 99.35%, showcasing the model's ability to accurately identify negative instances. However, there is a slight opportunity for enhancement in reducing false negatives, as indicated by the false negative rate (FNR) of 7.78%, which suggests that a small proportion of positive instances are being incorrectly taken as negative.

Fig. 5 depicts the training and testing loss of a model over a series of epochs. The graph shows two lines: a training loss and a testing loss. Both lines exhibit a downward trend as the number of epochs increases, suggesting that the model is learning and improving over time. The training loss decreases more steadily, while the testing loss shows some variability, which is common as the model tries to generalize from the training data to unseen data. This graph is a typical representation of a model's learning process in which the goal is to minimize loss, indicating better performance.

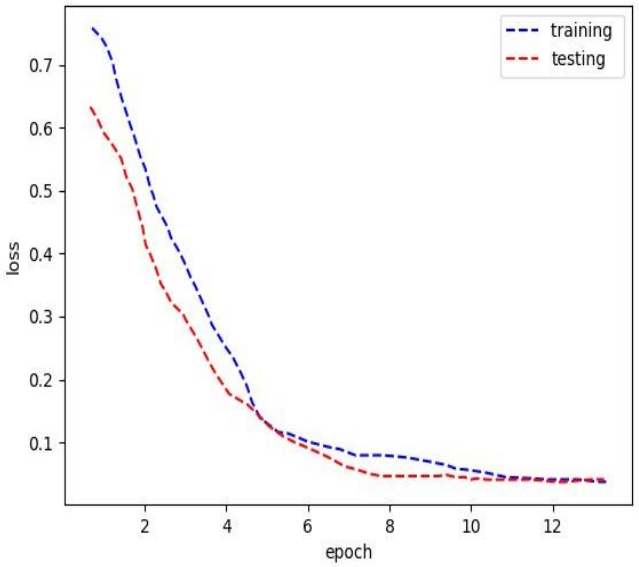

Fig. 5. Training and testing loss of a model.

Fig. 6 indicates the confusion matrix shows the performance of an object detection model across various classes of flowers, which is used to evaluate the performance of an object detection model on a set of test data for which the true values are known. The values of FPR (False Positive Rate) of 0.648702595 and FNR (False Negative Rate) of 7.784431138, suggest the model has a relatively high rate of incorrectly predicting the positive class and a low rate of missing actual positives.

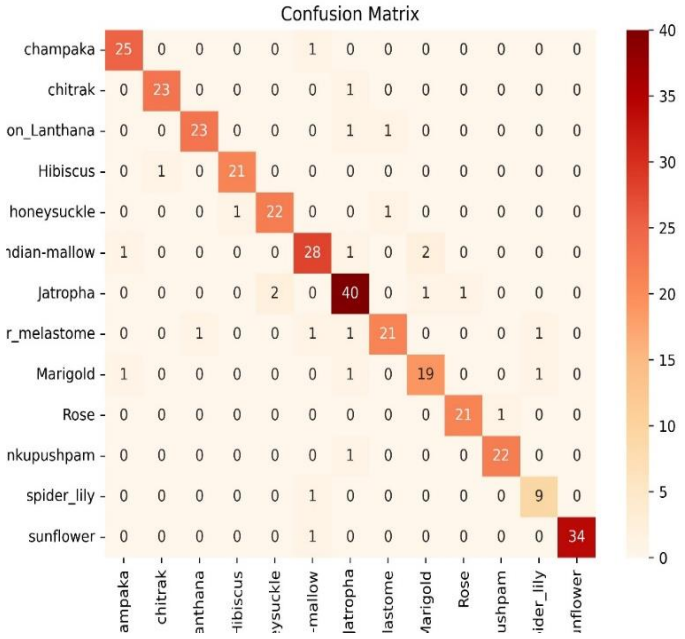

Fig. 6. Confusion matrix.

C. Comparative Analysis of Proposed Algorithms with Existing Algorithms

1) Comparative analysis of edge detection: Table II indicates the edge detection times of various algorithms were compared. The proposed GI-CED algorithm demonstrated the fastest edge detection with a processing time of approximately 7 ms, highlighting its superior computational efficiency. In contrast, the existing algorithms such as, Canny, Sobel, and Laplacian edge detection methods exhibited longer processing times of 8 ms, 12 ms, and 13 ms, respectively. The superior performance of the proposed GI-CED can be attributed to its innovative integration of gradient intensity enhancement with the Canny edge operator. This combination accelerates the edge detection process and improves the precision and clarity of the detected edges, making GI-CED particularly advantageous for real-time applications.

TABLE II. COMPARISON OF EDGE DETECTION TIME WITH VARIOUS ALGORITHMS

Methods	Time (ms)
Proposed GI-CED	7
Canny edge detection	8
Sobel edge detection	12
Laplacian edge detection	13

2) Comparative analysis of labelling: Comparison was done between proposed ZHA-KNN with existing algorithms.

Fig. 7 presents a comparison graph of clustering efficiency among various models. The proposed ZSA-KNN model achieves an impressive clustering efficiency of 94%, surpassing alternative methods such as KNN, Fuzzy Clustering, and Self-labelling, which achieve 87%, 90%, and 93%, respectively. This superior performance can be attributed to the ZSA-based distance measured-KNN (ZSA-KNN) algorithm utilized in the proposed model. The ZSA-KNN algorithm incorporates innovative distance measurement techniques, enabling more accurate clustering results and enhancing cluster efficiency. These improvements aim to elevate the clustering performance of ZSA-KNN, solidifying its position as a leading approach for efficient clustering in various domains.

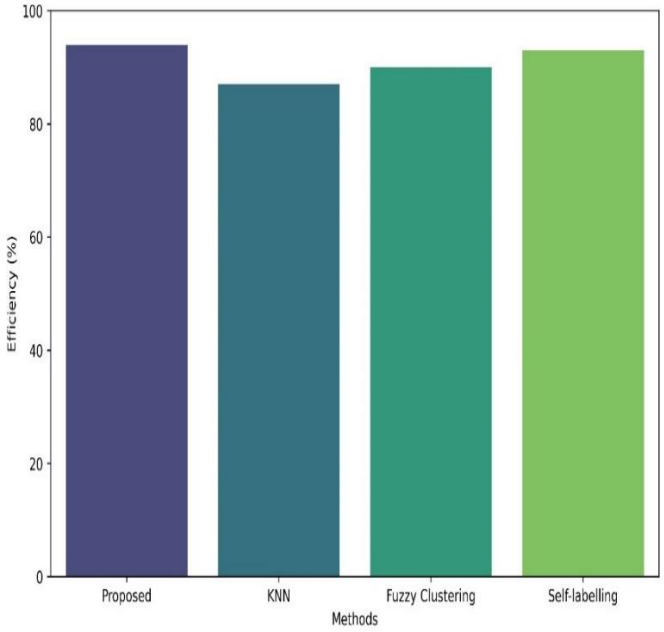

Fig. 7. Comparative graph of clustering efficiency among proposed vs various models.

TABLE III. COMPARISON OF PROPOSED HAMMING LOSS WITH VARIOUS ALGORITHMS

Methods	Hamming loss
Proposed ZSA-KNN	0.184
KNN	0.197
Fuzzy Clustering	0.233
Self-labelling	0.321

Table III presents a comparison of their Hamming Loss It is a metric used in multi-label object detection tasks to evaluate the fraction of incorrect labels to the total number of labels. It measures how many times an instance's predicted labels differ from the actual labels, with a lower value indicating better performance. The proposed ZSA-KNN method demonstrates a Hamming loss of 0.184, which is lower compared to other models such as KNN (0.197), Fuzzy Clustering (0.233), and Self-labelling (0.321). This indicates that the proposed ZSA-KNN method outperforms the other models, offering more accurate label predictions and thus better overall performance in multi-label object detection tasks.

Fig. 8 illustrates a comparison graph of clustering time (in milliseconds) among different methods. The proposed method, ZSA-KNN, demonstrates the lowest clustering time of 18ms, surpassing alternative approaches such as KNN, Fuzzy Clustering, and Self-labelling, which have clustering times of 21ms, 19ms, and 14.3ms, respectively. Overall, the lower

clustering time of the proposed ZSA-KNN method signifies a notable advancement in clustering efficiency, strengthening its potential for various data clustering applications.

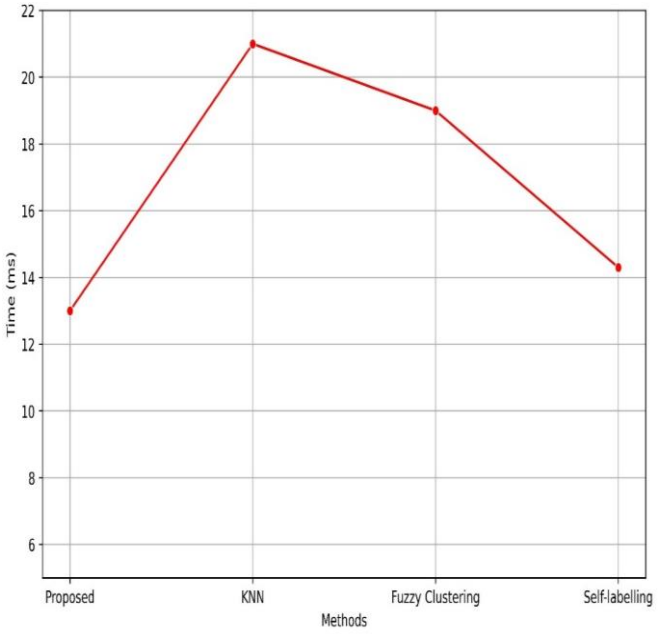

Fig. 8. Comparative graph of proposed clustering time (ms) vs different methods.

Fig. 9 depicts a comparison graph of one error and average precision for the proposed ZSA-KNN method compared to other methods. A lower one-error value indicates better performance, as it signifies fewer instances where the labelling top prediction is incorrect. Average precision calculates the area under the precision-recall curve, providing a measure of the labelling ability to rank relevant instances higher than irrelevant ones across all possible recall levels. The proposed ZSA-KNN method achieves a one-error of 0.226 and an average precision of 0.786, outperforming alternative methods such as KNN, Fuzzy Clustering, and Self-labelling, which exhibit one-error values of 0.239, 0.241, and 0.281, respectively, and average precision values of 0.761, 0.756, and 0.755, respectively.

3) Comparison of proposed object detection model with existing ones: Table IV presents a comparison of their respective accuracy rates. The proposed YoloV9 method achieves an impressive accuracy of 98.8%, significantly outperforming other models such as YoloV5, Rape Net, and YoloV5 with pruning, and Efficient Det, which demonstrate accuracies of 96.97%, 90.26%, 72%, and 94.36%, respectively. The innovative enhancements in the proposed YoloV9 ensure it stands out among existing algorithms, offering a solid solution for high-accuracy object detection needs in various applications.

Fig. 10 illustrates a comparison graph showcasing the segmentation performance of Mish-YoloV9 models with other methods. The proposed Mish-YoloV9 model exhibits outstanding performance with an accuracy of 98.8%, precision and recall of 92.21%, and an F1-score of 96.2%. In contrast, the standard YoloV9 model achieves an accuracy of 98%, precision of 97.7%, recall of 97.2%, and an F1-score of 98.1%. Additionally, Mish-YoloV9 outperforms other methods such as RNN and Res Net, showcasing its superiority. The enhanced performance of Mish-YoloV9 can be attributed to the utilization of the Mish activation function, which facilitates better feature extraction and pattern recognition capabilities within the network. This improvement underscores the efficacy of Mish-YoloV9 for segmentation tasks, highlighting its potential for various applications. Compared to alternative methods, the proposed model demonstrates notable advancements in capturing intricate data patterns and achieving superior segmentation results.

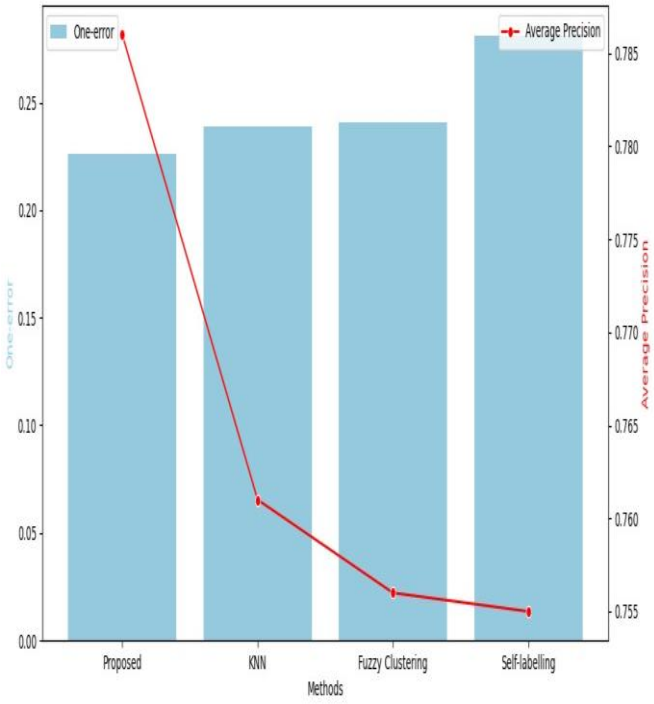

Fig. 9. Comparative analysis of labeling output.

TABLE IV. COMPARISON OF PROPOSED ACCURACY WITH VARIOUS ALGORITHMS

Methods	Accuracy (%)
Proposed YoloV9	98.8
YoloV5 [16]	96.97
Rape Net [17]	90.26
YoloV5 with prune [18]	72
Efficient Det [19]	94.36

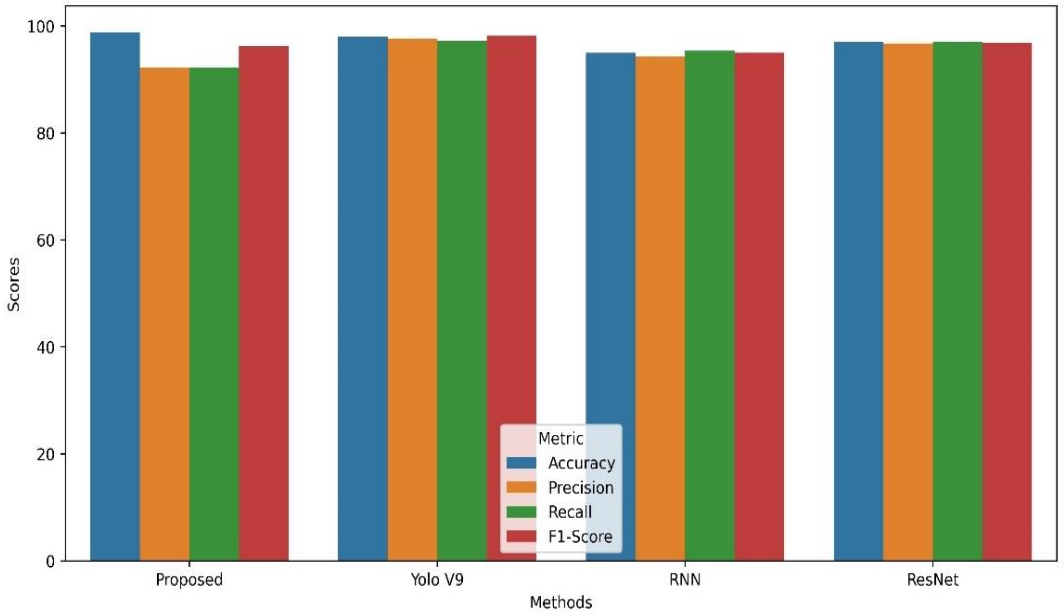

Fig. 10. Comparative segmentation performance of proposed models with other methods.

V. Conclusion

In this study, a novel flower-counting model has been proposed. This study considers images of 'Common Lanthana', 'Hibiscus', 'Jatropha', 'Marigold', 'Rose', 'champaka', 'chitrak', 'honeysuckle', 'Indian mallow', 'Malabar milestone', 'shanku pushpam', 'spider lily', 'sunflower'. These images are converted to a grayscale, then the noise is removed using a bilateral filter. Grayscale-converted images are then edge-detected using GI-CED, which obtained 7ms as edge detection time, likewise, edge-detected images are then given to labeling. Labeling was performed using the ZHA-KNN algorithm. ZHA-KNN obtained 18ms as clustering time and 94% clustering efficiency. Further, labeled images are then feature extracted, and features such as Convex Hull, Local Binary Patterns, Scale-Invariant Feature Transform, Hu, and Aspect Ratio are obtained. Extracted features along with the original image were given to MA-YoloV9. MA-YoloV9, obtained 98.8% accuracy with 4.56 as RMAE. Overall, the proposed model obtained better results than the previous model and was also able to count different flower images. This will be useful for flower crops harvest prediction and in importing flowers. Still, this model does not focus on different flowers in same image. In future, works can be developed to count different flowers in the same image using deep learning-based object detection model.

Statements and Declarations

A. Author Contributions

All authors contributed to the conception of the problem setting and overall design of the work. A.J, S.V, J.G and B.N built the conceptualization and methodology, A.J and S.V implemented the work, Validation was performed by A.J and J.G, writing was done by A.J, S.V and B.N. This version was revised and improved by all authors, who also read and approved the final manuscript.

B. Funding

No funding was received for conducting this study.

C. Conflict of Interest

The authors declare that they have no conflict of interest.

D. Ethical Approval

The research is original and all the figures and tables are created by the authors of this manuscript.

E. Consent to Participate

Not applicable.

F. Consent for Publication

All authors agree with the submission of the manuscript to this journal and possible publication afterwards.

References

[1] Peron, G., Franceschi, C., Da Dalt, C., Ferrarese, I., Sut, S., & Dall'Acqua, S. (2024). Biostimulation of Calendula officinalis with a soy protein hydrolysate induces flower and plant biomass and flower count by reversibly altering the floral metabolome. *Industrial Crops and Products*, *214*. https://doi.org/10.1016/j.indcrop.2024.118508.

[2] Estrada, J. S., Vasconez, J. P., Fu, L., & Cheein, F. A. (2024). Deep Learning based flower detection and counting in highly populated images: A peach grove case study. *Journal of Agriculture and Food Research*, *15*. https://doi.org/10.1016/j.jafr.2023.100930.

[3] Khokher, M. R., Liao, Q., Smith, A. L., Sun, C., MacKenzie, D., Thomas, M. R., Wang, D., & Edwards, E. J. (2023). Early Yield Estimation in Viticulture Based on Grapevine Inflorescence Detection and Counting in Videos. *IEEE Access*, *11*, 37790–37808. https://doi.org/10.1109/ACCESS.2023.3263238.

[4] Mann, G. S., Dubey, R. K., Singh, S., Deepika, R., Singh, D., & Kaur, N. (2023). Effect of growing media on growth and flowering of potted marigold (Tagetes erecta L.) irrigated with treated sewage water. *Journal of Plant Nutrition*, *46*(16), 4019–4032. https://doi.org/10.1080/01904167.2023.2220727.

[5] Valicharla, S. K., Wang, J., Li, X., Gururajan, S., Karimzadeh, R., & Park, Y. L. (2024). Morning Glory Flower Detection in Aerial Images Using Semi-Supervised Segmentation with Gaussian Mixture Models.

AgriEngineering, *6*(1), 555–573. https://doi.org/10.3390/agriengineering6010034.

[6] Herrera, D., Escudero-Villa, P., Cárdenas, E., Ortiz, M., & Varela-Aldás, J. (2024). Combining Image Classification and Unmanned Aerial Vehicles to Estimate the State of Explorer Roses. *AgriEngineering*, *6*(2), 1008–1021. https://doi.org/10.3390/agriengineering6020058.

[7] Yuan, W., Hua, W., Heinemann, P. H., & He, L. (2023). UAV Photogrammetry-Based Apple Orchard Blossom Density Estimation and Mapping. *Horticulturae*, *9*(2). https://doi.org/10.3390/horticulturae9020266.

[8] Houtman, W., Siagkris-Lekkos, A., Bos, D. J. M., van den Heuvel, B. J. P., Boer, M. den, Elfring, J., & van de Molengraft, M. J. G. (2021). Automated flower counting from partial detections: Multiple hypothesis tracking with a connected-flower plant model. *Computers and Electronics in Agriculture*, *188*. https://doi.org/10.1016/j.compag.2021.106346.

[9] Lin, J., Li, J., Ma, Z., Li, C., Huang, G., & Lu, H. (2024). A Framework for Single Panicle Litchi Flowers Counting by Regression with Multitask Learning. *Plant Phenomics*. https://doi.org/10.34133/plantphenomics.0172.

[10] Zhao, P., & Shin, B.-C. (2023). COUNTING OF FLOWERS BASED ON K-MEANS CLUSTERING AND WATERSHED SEGMENTATION. *J. Korean Soc. Ind. Appl. Math*, *27*(2), 146–159. https://doi.org/10.12941/jksiam.2023.27.146.

[11] Zhou, X., Sun, G., Xu, N., Zhang, X., Cai, J., Yuan, Y., & Huang, Y. (2023). A Method of Modern Standardized Apple Orchard Flowering Monitoring Based on S-YOLO. *Agriculture (Switzerland)*, *13*(2). https://doi.org/10.3390/agriculture13020380.

[12] Shinoda, R., Motoki, K., Hara, K., Kataoka, H., Nakano, R., Nakazaki, T., & Noguchi, R. (2023). RoseTracker: A system for automated rose growth monitoring. *Smart Agricultural Technology*, *5*. https://doi.org/10.1016/j.atech.2023.100271.

[13] Lamour, J., Le Moguédec, G., Naud, O., Lechaudel, M., Taylor, J., & Tisseyre, B. (2021). Evaluating the drivers of banana flowering cycle duration using a stochastic model and on farm production data. *Precision Agriculture*, *22*(3), 873–896. https://doi.org/10.1007/s11119-020-09762-y.

[14] Egi, Y., Hajyzadeh, M., & Eyceyurt, E. (2022). Drone-Computer Communication Based Tomato Generative Organ Counting Model Using YOLO V5 and Deep-Sort. *Agriculture (Switzerland)*, *12*(9). https://doi.org/10.3390/agriculture12091290.

[15] Gallmann, J., Schüpbach, B., Jacot, K., Albrecht, M., Winizki, J., Kirchgessner, N., & Aasen, H. (2022). Flower Mapping in Grasslands with Drones and Deep Learning. *Frontiers in Plant Science*, *12*. https://doi.org/10.3389/fpls.2021.774965.

[16] Li, X., Wang, X., Ong, P., Yi, Z., Ding, L., & Han, C. (2023). Fast Recognition and Counting Method of Dragon Fruit Flowers and FruitsBased on Video Stream. *Sensors (Basel, Switzerland)*, *23*(20). https://doi.org/10.3390/s23208444.

[17] Li, J., Wang, E., Qiao, J., Li, Y., Li, L., Yao, J., & Liao, G. (2023). Automatic rape flower cluster counting method based on low-cost labelling and UAV-RGB images. *Plant Methods*, *19*(1). https://doi.org/10.1186/s13007-023-01017-x.

[18] Yu, G., Cai, R., Luo, Y., Hou, M., & Deng, R. (2024). A-pruning: a lightweight pineapple flower counting network based on filter pruning. *Complex and Intelligent Systems*, *10*(2), 2047–2066. https://doi.org/10.1007/s40747-023-01261-7.

[19] Zhu, R., Wang, X., Yan, Z., Qiao, Y., Tian, H., Hu, Z., Zhang, Z., Li, Y., Zhao, H., Xin, D., & Chen, Q. (2022). Exploring Soybean Flower and Pod Variation Patterns During Reproductive Period Based on Fusion Deep Learning. *Frontiers in Plant Science*, *13*. https://doi.org/10.3389/fpls.2022.922030.

[20] Kaur, R., Jain, A., & Kumar, S. (2021). Optimization classification of sunflower recognition through machine learning. *Materials Today: Proceedings*, *51*, 207–211. https://doi.org/10.1016/j.matpr.2021.05.182.

[21] Jiang, Y., Li, C., Xu, R., Sun, S., Robertson, J. S., & Paterson, A. H. (2020). DeepFlower: a deep learning-based approach to characterize flowering patterns of cotton plants in the field. *Plant Methods*, *16*(1). https://doi.org/10.1186/s13007-020-00698-y.

[22] Yang, Y., Zhang, G., Ma, S., Wang, Z., Liu, H., & Gu, S. (2024). Potted Phalaenopsis Grading: Precise Bloom and Bud Counting with the PA-YOLO Algorithm and Multiviewpoint Imaging. *Agronomy*, *14*(1). https://doi.org/10.3390/agronomy14010115.

[23] Vo, H.-T., Chau Mui, K., Nguyen Thien, N., & Pham Tien, P. (2024). Automating Tomato Ripeness Classification and Counting with YOLOv9. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 15, Issue 4). www.ijacsa.thesai.org.

[24] Shankarpure, M. R., & Patil, D. D. (n.d.). Smart Fruit Identification and Counting using Machine Vision Approach. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 14, Issue 12). www.ijacsa.thesai.org.

[25] Wang, C., Han, Q., Li, C., Li, J., Kong, D., Wang, F., & Zou, X. (2024). Assisting the Planning of Harvesting Plans for Large Strawberry Fields through Image-Processing Method Based on Deep Learning. *Agriculture*, *14*(4), 560. https://doi.org/10.3390/agriculture14040560.

[26] Patton, A. J., Higginbotham, R. A., Law, Q. D., & Weisenberger, D. V. (2022). Counting dandelion blooms in field plots using an image processing program. *International Turfgrass Society Research Journal*, *14*(1), 390–396. https://doi.org/10.1002/its2.32.

[27] Viveros Escamilla, L. D., Gómez-Espinosa, A., Escobedo Cabello, J. A., & Cantoral-Ceballos, J. A. (2024). Maturity Recognition and Fruit Counting for Sweet Peppers in Greenhouses Using Deep Learning Neural Networks. *Agriculture (Switzerland)*, *14*(3). https://doi.org/10.3390/agriculture14030331.

[28] Rahim, U. F., & Mineno, H. (2020). Tomato Flower Detection and Counting in Greenhouses Using Faster Region-Based Convolutional Neural Network. *Journal of Image and Graphics*, *8*(4), 107–113. https://doi.org/10.18178/joig.8.4.107-113.

[29] Ge, Y., Lin, S., Zhang, Y., Li, Z., Cheng, H., Dong, J., Shao, S., Zhang, J., Qi, X., & Wu, Z. (2022). Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. *Machines*, *10*(6). https://doi.org/10.3390/machines10060489.

[30] Wang, X. (Annie), Tang, J., & Whitty, M. (2021). DeepPhenology: Estimation of apple flower phenology distributions based on deep learning. *Computers and Electronics in Agriculture*, *185*. https://doi.org/10.1016/j.compag.2021.106123.