

Marigold Flower Blooming Stage Detection in Complex Scene Environment using Faster RCNN with Data Augmentation

Sanskriti Patel

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Charotar University of Science and Technology
Changa, India

Abstract—In recent years, flower growing has developed into a lucrative agricultural sector that provides employment and business opportunities for small and marginal growers in both urban and rural locations in India. One of the most often cultivated flowers for landscaping design is the Marigold flower. It is also widely used to create garlands for ceremonial and social occasions using loose flowers. Understanding the appropriate stage of harvesting for each plant species is essential to ensuring the quality of the flowers after they have been picked. It has been demonstrated that human assessors consistently used a category scoring system to evaluate various flowering stages. Deep learning and convolutional neural networks have the potential to revolutionize agriculture by enabling efficient analysis of large-scale data. In order to address the problem of Marigold flower stages detection and classification in complex real-time field scenarios, this study proposes a fine-tuned Faster RCNN with ResNet50 network coupled with data augmentation. Faster RCNN is a popular deep learning framework for object detection that uses a region proposal network to efficiently identify object locations and features in an image. The Marigold flower dataset was collected from three different Marigold fields in the Anand District of Gujarat State, India. The collection includes photos that were taken outdoors in natural light at various heights, angles, and distances. We have developed and fine-tuned a Faster RCNN detection and classification model to be particularly sensitive to Marigold flowers, and we have compared the generated method's performance to that of other cutting-edge models to determine its accuracy and effectiveness.

Keywords—Deep learning; convolutional neural networks; object detection; marigold flower blooming stage detection

I. INTRODUCTION

One of the main economic pillars in India is agriculture. For roughly 58% of Indians, agriculture is their main source of income. The field of horticulture known as "flower farming," also referred to as "floriculture," deals with the study of cultivating and selling flowers and foliage plants. It primarily focuses on growing ornamental plants, cultivated greens, potted flowering plants, tubers, rooted cuttings, cut flowers, and other floriculture products. In recent years, flower farming has become a successful agriculture industry that offers employment and entrepreneurship prospects in both urban and rural areas, as well as for small and marginal farmers [1]. One of the most often cultivated flowers for garden ornamentation is the Marigold, which is also widely used as loose flowers to

create garlands for ceremonial and social occasions. The Marigold is one of the most popularly grown flowers for landscape decoration. It is also frequently used as loose flowers to make garlands for ceremonial and social occasions. Marigold is mostly used to treat various skin disorders, such as varicose veins, contusions, and bruises. Additionally, inflammation and minor skin wounds can be successfully addressed. Marigold cream aids in the healing of sunburns and eczema wounds. Marigold farming is a profitable activity that requires little maintenance and effort. Marigold cultivation is a profitable activity as it requires less investment and gives better harvest with a high profit [2].

To guarantee the quality of the flowers after harvest, it is crucial to understand the ideal stage of harvesting for each plant type. The flower's life is considerably decreased when it is harvested too early or too late. A flower normally becomes larger as it progresses from bud to bloom. A flower like a daisy, Marigolds can only be picked when completely opened [3].

Identifying the plant flowering status have traditionally needed human evaluators to manually inspect flower fields and report flowering status. It has been shown that human assessors regularly assessed different flowering stages using a category scoring system. For instance, you could want to know when 30% of the flowering plants in a field have blooms that are fully open. This made it possible for researchers to compute the time between various blooming phases [4]. Deep learning advancements and innovations make it possible to quickly characterize the flowering patterns of field-grown plants. It is frequently necessary to regularly spot and count newly opening blooms on plants when cultivating flowers like Marigold.

Marigold farming is a profitable activity as it requires less investment and gives better harvest with a high profit. To guarantee the quality of the flowers after harvest, it is important to understand the ideal stage of harvesting for each plant type. Deep learning advancements and innovations make it possible to quickly characterize the flowering patterns of field-grown plants. Using a cutting-edge object detector called Faster Region-based Convolutional Neural Network, we propose an efficient method to detect and classify Marigold flowers of various stages in diverse field conditions. The proposed method is inspired by successful studies using deep

Convolutional Neural Networks (CNNs) in difficult computer vision and object detection tasks.

This study suggests a Faster RCNN network coupled with image augmentation to address the challenge of Marigold flower stages identification and classification in complicated real-time field situations. In order to detect and classify emerging blooms of Marigold flower plants, the objectives of the study are: (1) To acquire and pre-process images of Marigold flowers in challenging real-time environments; (2) To develop and tune a Faster RCNN detection and classification model to become particularly sensitive to Marigold flowers; and (3) To assess the accuracy and efficacy of the developed approach by comparing its performance to that of other cutting-edge models.

II. RELATED WORK

D. Thi Phuong Chung and D. Van Tai [5] represent a deep learning based technique for fruit detection. They have EfficientNet architecture that recognized fruit objects from the Fruit 360 dataset and achieved 95% accuracy. A. Rocha et al. [6] introduced a novel method for classification of fruit and vegetables from images. A multi-class fruit-and-vegetable categorization task in a semi-controlled setting, like a distribution centre or the supermarket checkout line, is used to validate the newly presented fusion approach. According to the findings, the solution can lower the classification error by up to 15% compared to the baseline. I. Sa et al. [7] present a novel approach to fruit detection using deep convolutional neural networks. The goal is to develop a fruit detection system as it is a critical component of an autonomous agricultural robotic platform and is essential for estimating fruit production and automating harvesting. They have proposed a multi-modal Faster RCNN model that, when compared to earlier work, delivers state-of-the-art results, with F1 score performance for the detection of sweet pepper increasing from 0.807 to 0.838.

Moreover, T. Abbas et al. [8] mentioned different smartphone applications like LeafSnap [9], and Pl@ntNet [10], that can be used to identify flowers rapidly. I. Patel and S. Patel [11] proposed an optimized deep learning model that detects the flower species. For that, they have integrated Faster RCNN with Neural Network Search and Feature Pyramid. The mAP score obtained on the standard Oxford flower species dataset is 87.6%. D. Wu et al. [12] proposed a methodology for detecting *Camellia oleifera* Fruit using YOLOv7 object detection model. For the research, they have collected the dataset from the complex scene and applied different evaluation metrics. The values derived for mAP is 96.03%, Precision is 94.76%, Recall is 95.54%, and F1 score is 95.15%. S. Nuanmeesri et al. [13] proposed a novel method that predicts disease from Marigold flower images. The outcome demonstrated that the model created using the watershed dataset is the most effective. The model's validation accuracy was 88.03%, validation loss was 4.21%, and model testing accuracy was 91.67%.

From the literature survey, we have found that deep learning algorithms and models can be applied for detecting objects from real-time environment and it has a great significance and theoretical value. Moreover, there is a need for an automated model that provides improved accuracy and generalization across different growing conditions and

environmental factors. There are challenges existed in developing models that can handle the variability in marigold flower appearance across different growing conditions, such as varying lighting, background, or growth stages. This research aimed to develop and propose a more sophisticated model that can handle these challenges and improve the accuracy and generalization of marigold flower blooming stage identification and classification.

III. MATERIALS AND METHODS

A. Acquisition and Pre-Processing of Marigold Flower Images in Complex Scene Environment

One Canon DSLR Dual Lens Camera and two smartphones were used to capture images of Marigold flowers in the Marigold cultivation agricultural fields under natural daylight illumination. Three distinct agricultural Marigold fields in the Anand District of Gujarat, India, were chosen for the study. Three regular harvesting times in the winter November and December months have chosen for the capturing of images. The collection consists of 550 photos in total, each with a resolution of 4000 by 2250 pixels and taken at various heights, angles, and random distances in natural light conditions. The dataset was captured in two stages. The acquired images are having different conditions like top angled, side angled, heavily occluded, lightly occluded, overlapped, etc. are represented in the Fig. 1.

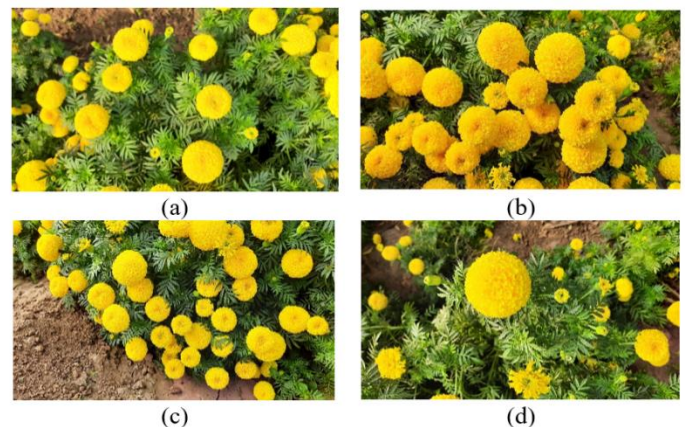


Fig. 1. Examples of Marigold images taken in various settings (a) Occlusion in image, (b) Overlapped marigold flowers and buds, (c) Image captured from side angle (d) Image captured from top angle.

Because they provide the supervised learning algorithm with the training data, image annotations play a key role in computer vision algorithms. Using the graphical image annotation tool namely, LabelImg [14], which is built on Python, the complete dataset was annotated and saved in XML documents. The images are annotated in two classes; bud and flower which are represents and differentiates their growing stages.

B. Data Augmentation

Data augmentation refers to the process of artificially expanding the size of a training dataset by creating modified versions of images in the dataset [15]. This can be useful in object detection, especially for Faster RCNN, as it helps prevent overfitting and can improve the model's ability to

generalize to new, unseen data. The most popular technique of fundamental augmentation is geometric transformation [16]. The transformation's parameters may be preset or chosen at random. In this research, the common techniques for data augmentation used are flipping, scaling and rotating. Flipping is the process used to rotate an image either along its vertical axis or along its horizontal axis. In contrast to vertical flipping, which flips the image on the vertical axis, horizontal flipping flips the image on the horizontal axis [17]. An image can be rotated by adding a rotational angle. The image is rotated in a random direction to produce enhanced images. Here, the left and right axes of rotation are chosen at random. The dataset includes an image that can be zoomed in or out. One of the most popular data augmentation techniques is zooming. It is possible to conduct a zoom between 0.5% and 1.0%.

After the data augmentation applied, the final augmented training set consists of 1583 images that help to improve the generalization ability of the detection model and avoid the overfitting of the detection model.

IV. PROPOSED MODEL FOR MARIGOLD FLOWER BLOOMING STAGE DETECTION IN COMPLEX SCENE ENVIRONMENT USING FASTER RCNN WITH DATA AUGMENTATION

The Fig. 2 represents the stepwise architecture of the proposed model for Marigold flower stages identification and classification.

There are two main types of object detection models: one-stage models and two-stage models. One-stage object detection models, also known as single-shot detectors, are designed to detect objects in a single pass over an input image [18]. These models use a single neural network to simultaneously predict object locations and class labels. Two-stage object detection models, on the other hand, are designed to detect objects in a two-step process. The first step involves generating a set of potential object locations, known as region proposals, using a separate network called a region proposal network (RPN). The second step involves classifying the region proposals and refining their locations using another network. Two-stage models typically achieve higher accuracy than one-stage models, but at the cost of slower inference speed. One-stage models, on the other hand, are faster but can be less accurate, especially for small objects or objects with high aspect ratios [19] [20]. This research is mainly focuses on identifying two stages of Marigold flower growth; fully grown flower and a bud. Bud is a small flower object that is to be detected by the proposed model. Therefore, in this research we have proposed a two-stage object detection model i.e. Faster RCNN.

We are primarily interested in object detection in our study because it is the first step in determining whether a flower is a bud or a fully blown blossom. So, using Faster RCNN, we simulate a particular generic detector. In order to create an effective technique for looking for instances of flowers and buds in a flower image, we make use of the object suggestions trained by RPN and their associated features derived from a ResNet50 CNN architecture. By combining the convolutional strengths of RPN and Fast RCNN utilising the present neural network formulation, we further combine RPN and Fast RCNN into a coherent model. The feature network, RPN, and detection network are the three deep networks that make up the

suggested methodology. A boxing approach used by faster RCNN enables the operator to specify the potential regions that will be introduced into the RPN. With the suggested approach, we begin by performing a CNN model using our dataset of Marigold flowers. After examining the input image, a selective search procedure is then used to extract a region of interest (RoI). The candidates between the closest raster frames are then refined using the prepared deep model to classify the extracted ROIs into candidates.

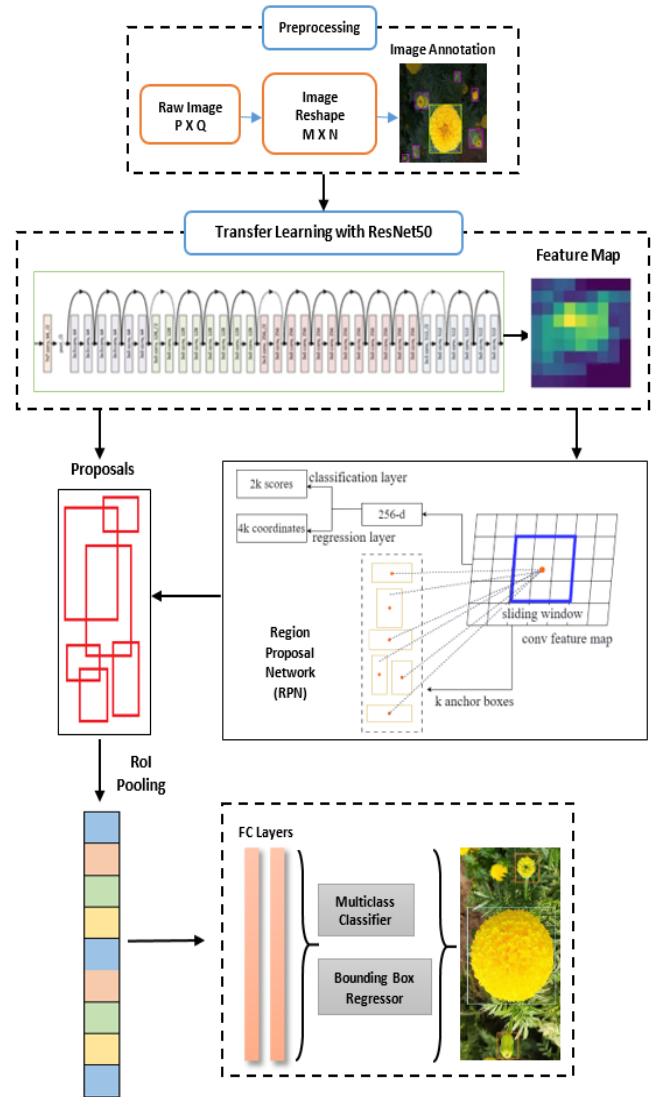


Fig. 2. A proposed two-stage object detection model for marigold flower blooming stage detection in complex scene environment using faster RCNN with data augmentation.

A. Image Pre-Processing and Annotation

In object detection, pre-processing refers to the steps taken to prepare the input data for the object detection model. This may include tasks such as resizing the image, normalizing pixel values, converting the image to grayscale, etc. [21]. In this research, image is reshaped and annotated with two labels i.e. flower and bud. The dataset is splitted into training and validation sets by having a ratio of 90:10. No repeated images

among the training, validation and test sets were ensured to prevent overfitting of the model [22] [23].

Image annotation in object detection refers to the process of labeling objects within an image to train a machine learning model for object detection. The annotated data is used to train the model to detect and classify objects within new, unseen images [24]. Image annotation involves drawing bounding boxes around objects of interest and assigning a label to each bounding box. The goal of image annotation is to provide the model with enough data to learn the features and characteristics of different objects, so it can accurately detect them in new images [25]. LabelImg is a graphical image annotation tool that is used to label images for object detection in machine learning. The tool provides an interface for drawing bounding boxes around objects in an image and assigning class labels to the objects. The resulting annotations are saved in an XML file that can be used as input for training machine learning models. The Fig. 1 illustrates the image annotation performed using LabelImg tool. As mentioned earlier, the images are annotated in two classes; bud and flower which are represents and differentiates their growing stages.

Here, a 2D bounding boxes annotation for flower species detection is applied, as illustrated in Fig. 3. The 2D bounding boxes are applied by drawing rectangles or cuboids around flower objects in an image, and then, labels of respective flower classes are applied to them.

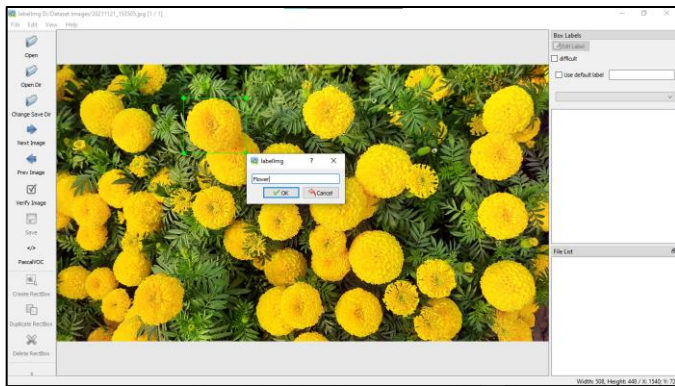


Fig. 3. Image annotation using labelImg.

B. Faster RCNN with ResNet50

Faster RCNN is a popular object detection architecture that is used for Marigold flower stages detection. Faster RCNN is a state-of-the-art object detection algorithm that combines the two-stage object detection framework with deep convolutional neural networks [26]. The first stage is a region proposal network (RPN) that generates a set of candidate object regions. These regions are then fed into the second stage, which is a Fast RCNN network that classifies the regions and refines the bounding box locations. The two stages work together to efficiently detect objects in an image by first proposing a large number of potential regions and then using the Fast RCNN network to accurately classify and locate the objects. The key advantage of Faster RCNN is its end-to-end training, which enables it to learn to detect objects directly from image data without relying on heuristics or manually-designed features [27].

The proposed Faster RCNN model with ResNet50 for flower object detection can be divided into the following modules:

1) *ResNet50 backbone*: This module consists of the pre-trained ResNet50 network that serves as the feature extractor. It takes an image as input and outputs a feature map [28]. In Faster RCNN, the ResNet-50 backbone is used as the feature extractor network to produce a compact feature representation of the input image [29]; the input image being fed into the backbone convolutional neural network. For that, the input image is first resized by considering the shortest px with the longer side not exceeding 1000px. The output of the backbone network is a feature map. These feature maps are then fed into the Fast RCNN network for classification and bounding box regression. The use of a pre-trained ResNet50 network as the feature extractor allows Faster RCNN to leverage the information learned from the large-scale image classification task, improving its object detection performance. Additionally, the use of ResNet50 as a backbone allows for transfer learning, where the feature extractor can be fine-tuned for the specific object detection task using a smaller dataset [30].

With the primary goal of resolving the vanishing/exploding gradient issue, ResNet architecture established the Residual Network concept. The network employs a method known as "skip connection" for that. The ResNet architecture is known for its use of "skip connections" or "shortcut connections" [31]. These skip connections help alleviate the problem of vanishing gradients in very deep neural networks. Skip connections in ResNet work by allowing the network to bypass one or more layers, effectively allowing the gradients to be backpropagated directly to earlier layers as illustrated in Fig. 4. This helps to preserve the information from the original input, making it easier for the network to learn and improve.

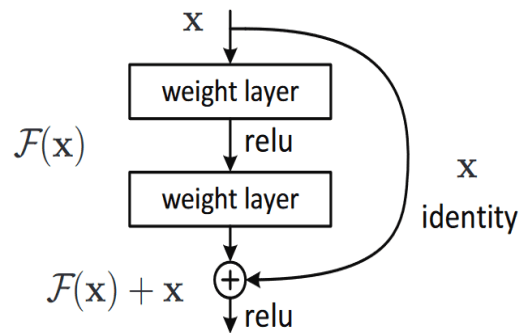


Fig. 4. An illustration of skip connection [32].

Without using skip connection, input 'x', multiplied by the layer's weights, followed by adding a bias term:

$$H(x) = f(wx + b) \quad (1)$$

or

$$H(x) = f(x) \quad (2)$$

With the introduction of skip connection, the output of the layer changes to

$$H(x) = f(x) + x \quad (3)$$

The loss function used in ResNet50, like most of deep learning models, is typically a categorical cross-entropy loss. This loss measures the dissimilarity between the predicted class probabilities and the true class label, and is commonly used for multi-class classification problems. The following is an equation to calculate the categorical cross-entropy [33].

$$\text{Cross - entropy} = -\frac{1}{N} \sum_{i=1}^N \log P_{\text{model}}[y_i \in C_{y_i}] \quad (4)$$

2) *Region Proposal Network (RPN)*: The Region Proposal Network (RPN) is a crucial component of the Faster RCNN object detection framework. Its primary responsibility is to generate a set of candidate object regions in the input image, called region proposals. A region proposal network generates several regional proposals [34]. These proposals submit to the identification network's detection. The three components of RPN are the anchor window, loss function, and set of region proposals [35]. RPN adopts the sliding window methodology because a small sub-network is evaluated on a dense 3x3 sliding window in the RPN design. The IoU ratios and the ground-truth bounding boxes can thus be used by the RPN to produce numerous anchors [36].

The RPN uses anchor boxes, which are predefined bounding box shapes, to guide the generation of the region proposals. The network outputs are then combined with the anchor boxes to produce the final set of region proposals. The step-wise process is described as follows [37]: (i) RPN utilize a sliding window for each region over the feature map. (ii) To generate region proposals, k (k=9) anchor boxes are employed for each site, with 3 scales of 128, 256, and 512 and 3 aspect ratios of 1:1, 1:2, and 2:1. (iii) Whether an object is present or not, a CLS layer produces 2k scores for k boxes. (iv) For the box centre coordinates, width, and height of k boxes, a reg layer outputs 4k. (v) There are WHk anchors overall with a WH feature map size.

The total loss of the RPN is calculated by the multitask loss function. The calculation formula is [38].

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (5)$$

Where N_{cls} represents the number of batch training data, N_{reg} represents the number of anchors, λ represents the balance weight.

$L_{cls}(p_i, p_i^*)$ is the logarithmic loss function defined as;

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (6)$$

$p_i^* L_{reg}(t_i, t_i^*)$ is the regression loss calculated by the following Smooth L1 function:

$$L_{reg}(t_i, t_i^*) = \begin{cases} 0.5 (t_i - t_i^*)^2, & |x| < 1 \\ |t_i - t_i^*| - 0.5, & \text{otherwise} \end{cases} \quad (7)$$

Where p_i is the probability of the anchor being predicted as the target, and p_i^* is the truth value of the prediction result: if the anchor is predicted as a positive sample, the value of tag p_i^* is 1; otherwise, the value is 0; $t_i = \{t_x, t_y, t_w, t_h\}$ is

the location of the predicted detection box; and t_i^* is the ground truth coordinate.

RPN network must therefore check in advance which location contains the object. The detection network will then receive the relevant locations and bounding boxes and use them to identify the object class and deliver the object's bounding box.

3) *RoI pooling layer*: RoI (Region of Interest) pooling is a technique used in Faster RCNN for processing the region proposals generated by the RPN [39]. RoI pooling is a layer in the Fast RCNN network that takes as input the feature map produced by the ResNet-50 backbone and a set of region proposals. The RoI pooling layer resizes each region proposal to a fixed size, regardless of its original size or aspect ratio, and aggregates the features within each region into a compact feature representation [40]. This enables the Fast RCNN network to perform classification and regression on the objects in the image, regardless of their size and aspect ratio. RoI pooling is critical for Faster RCNN's ability to accurately detect and classify objects of different sizes and aspect ratios in an image. The RoI pooling layer allows the Fast RCNN network to have a fixed input size, making it easier to train and optimize, while still allowing it to handle objects of varying sizes in the image.

4) *Fast RCNN classifier and bounding box regressor*: In Faster RCNN, after the RoI pooling layer, the features of the region proposals are fed into the classifier and bounding box regressor [41]. The classifier is a fully connected layer that performs object classification by predicting the probability of each region proposal belonging to each of the predefined object classes. The classifier outputs a score for each region proposal and class, indicating the likelihood of the presence of an object of that class in the region. The bounding box regressor is another fully connected layer that performs bounding box regression [42]. It takes as input the feature representation of the region proposals and outputs the adjustments to the locations of the region proposals, refining their locations to better fit the objects in the image.

Together, the classifier and bounding box regressor form the Fast RCNN network, which accurately detects and classifies objects in the image by combining the information from the region proposals, the classifier scores, and the refined bounding box locations.

V. NETWORK TRAINING PLATFORM AND PARAMETER SETTINGS

The experiment is carried out using a machine having NVIDIA Tesla V100 32GB PCIe based GPU card equipped with 64GB RAM and Intel Xeon 6226R processor. It has 4 units of 6TB SATA 7200 RPM 3.5" HDD and it is running with Ubuntu 21.04 operating system. We have used TensorFlow 2 Object Detection API with CUDA 11.2, CuDNN 8.1.0 and Python 3.8 virtual environment.

An illustration of a training pipeline developed for experiment with numerous separate activities is shown in Fig. 5. The image annotation tool is used to build the labelled

flower datasets (i.e. LabelImg). All of the datasets for tagged flowers are saved as .csv files, which are then transformed into .record files and used as inputs by the networks to forecast bounding boxes and confidences for objects. TensorFlow's object identification model needs a Label Map that converts each of the applied labels into an integer value. Both the training and evaluation processes use this Label Map. Files with the ending ".pbtxt" are label map files. We have used ResNet50 as a pre-trained CNN and modified the Faster RCNN detection model that is trained using COCO 2017 dataset and made available by Tensorflow Object Detection API - Model Zoo [43]. Finally, loss functions are used to measure the accuracy of the training process and an inference graph are generated at the end of the training pipeline.

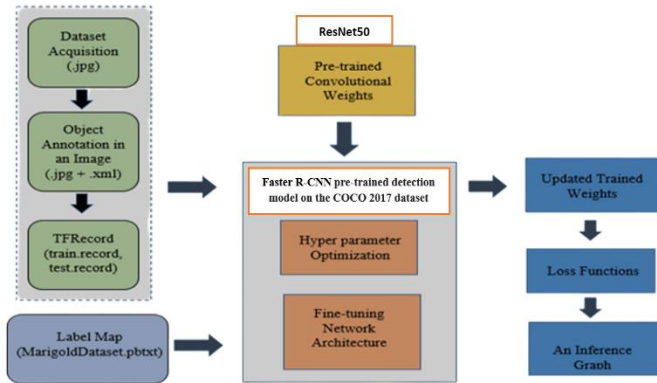


Fig. 5. A training pipeline for experiment.

A. Hyperparameters

The model parameter values that a learning algorithm ultimately learns are defined by hyperparameters, which are variables whose values have an impact on the learning process. The selection of hyperparameters that aid in an object identification model's optimum accuracy has an impact on the model's accuracy as well. Therefore, figuring out the best values for these factors is a challenging task [44]. Hyperparameter tuning, often known as optimization, is the process of selecting the best set of parameters for a model's learning procedure [45]. In this research, we have setup multiple hyperparameters such as learning rate, batch size, number of steps, activation function and dropout rate. The learning rate for the proposed model sets to 0.002, the batch size chose was 16. The input size was set to 640×640 . The training Epoch was set to 1000. During the training process, Tensorboard visualization tool was used to record data and observe loss, and save the model weight of every epoch.

B. Evaluation Indicators of Model

In this study, the model's performance was accurately and impartially assessed using Precision, Recall, Mean Average Precision (mAP), and F1 score. The number of correct targets divided by the total number of targets is known as the precision evaluation index [46]. The detection impact will generally be better the higher the Precision. Precision is a highly logical evaluation metric, however occasionally a high Precision score does not mean everything. Thus, mAP, Recall, and F1 score were developed for thorough examination.

$$\text{Precision} = \frac{TP}{TP+FP} \times 100 \quad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (9)$$

$$\text{Average Precision (AP)} = \int_1^0 P(r) dr \quad (10)$$

$$\text{Mean Average Precision (mAP)} = \frac{1}{n} \sum_{i=1}^n AP_i \quad (11)$$

Other than above evaluation indicators, IoU is also used. The amount of overlap between the predicted and ground truth bounding boxes is indicated by the IoU value, which ranges from 0 to 1 as described in Fig. 6 [47].

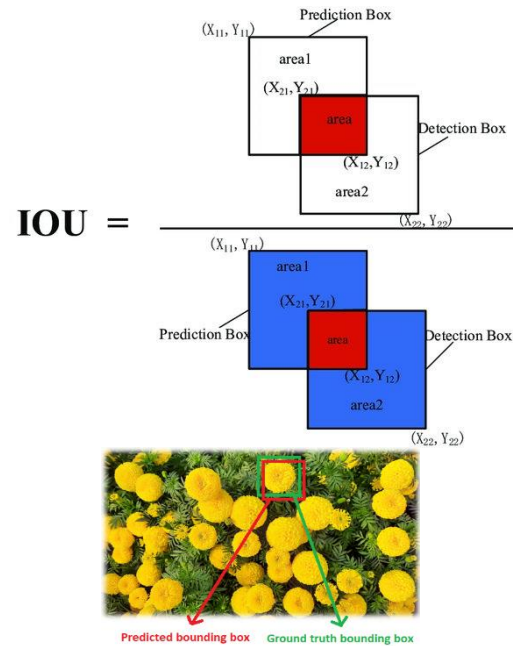


Fig. 6. A description of Intersection-over-Union (IoU)

There is no overlap between the boxes if the IoU is 0. When the union of the boxes equals their overlap and the IoU = 1, this signifies that the boxes are entirely overlapping. The equation for the same is illustrated as Eq. (3).

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (12)$$

VI. RESULTS AND DISCUSSION

To experiment, we have used TensorFlow Object Detection API. It is an open-source framework which is built on top of the TensorFlow library, offers a variety of pre-trained object detection models as well as tools for creating and training unique object detection models. The pre-trained models, also known as the Model Zoo, feature various models that are pre-trained on the COCO dataset, which is a large-scale object detection, segmentation, and captioning dataset. We have used the learned weights from these pre-trained model and fine-tuning these pre-trained models on our own Marigold datasets.

The proposed Faster RCNN with ResNet50 model is compared with one-stage object detection model that is SSD (Single Shot Detector). By examining the Faster RCNN of various networks, the training model and the subsequent

classification results of mean average precision are obtained and presented in Table I and Table II. We have experimented both of the object detection model with the original dataset and augmented dataset. On the original Marigold dataset, the proposed Faster RCNN with ResNet50 model provides 88.71% mAP score with 4.312 average detection speed per second. SSD MobileNet model obtained 74.30% mAP score which is lower as compare to the proposed Faster RCNN with ResNet50 model.

We have synthetically increases the size of the dataset and experimented both of the model. This time, the proposed Faster RCNN with ResNet50 model provides 89.47% mAP score and SSD MobileNet model obtains 78.12% mAP score. Although, it is observed that the average detection speed of SSD MobileNet is quite high as compare to the Faster RCNN with ResNet50.

TABLE I. RESULTS AND PERFORMANCE COMPARISON AMONG DIFFERENT DETECTION MODELS

Data Augmentation	Target Detection Networks	mAP (%)
Without Data Augmentation	Proposed Faster RCNN ResNet50 V1 640X640	88.71
	SSD MobileNet V1 FPN 640X640	74.30
With Data Augmentation	Proposed Faster RCNN ResNet50 V1 640X640	89.47
	SSD MobileNet V1 FPN 640X640	78.12

TABLE II. COMPARISON OF AVERAGE DETECTION SPEED AMONG DIFFERENT DETECTION MODELS

Target Detection Networks	Average Detection Speed (s/Image)
Proposed Faster RCNN ResNet50 V1 640X640	4.312
SSD MobileNet V1 FPN 640X640	0.64

Fig. 7 contains the examples of Intersection over Union (IoU) detection results of the proposed object detection on real-time Marigold Dataset captured in complex scene environment. The figure (a) and (b) are the images with detection results. It shows the detection of two classes; flower and bud. These two classes represent the flower growing stages that can be helpful to decide the harvesting time.



(b)

Fig. 7. Examples of Intersection over Union (IoU) detection results of the proposed object detection on real-time marigold dataset captured in complex scene environment. Fig. 7(a) and 7(b) are the images with detection of flower and bud objects

In conclusion, several studies have been conducted in the past to address the issue of marigold flower object identification and classification. Marigold flower blooming stage identification and classification using deep learning techniques such as Faster RCNN, YOLO, and SSD has gained significant attention in recent years. These techniques offer faster and more accurate detection and classification of marigold flower blooming stages. YOLO and SSD are faster, they are also less accurate than the two-stage models, particularly on small objects [48]. Several researchers proposed variety of methods for classification using the machine learning and deep learning algorithms that achieved good performance. However, they have considered image classification based on discrete features of flower image [49] [50][51][52]. In this research, we have proposed a method based on two-stage object detection. The proposed method allows for more precise localization and classification of objects and can be used to quickly and accurately identify the blooming stage of marigold flowers. Moreover, two-stage object detectors are more robust to variations in lighting conditions, background clutter, and occlusions compared to single-stage detectors. This makes them ideal for identifying marigold flowers blooming stages, which can be affected by different lighting conditions and may have complex backgrounds.

VII. CONCLUSION

In order to detect Marigold flower stages in intricate agricultural field settings, a real-time and accurate identification strategy based on a two-stage Faster RCNN object detection network with data augmentation was presented in this study. We have gathered and analysed data on Marigold flowers in a variety of field settings as part of our research. All of the dataset's images were divided into two classes: bud and flower. The flower growth stage is represented by these two classes. Geometric data augmentation techniques were also used to improve the dataset. Then, utilising the ResNet50 backbone network, we fine-tune the two-stage object detector, namely Faster RCNN. We conducted an experiment and compared the outcomes with SSD MobileNet, a single-stage object identification model. The findings suggest that data augmentation can significantly enhance the proposed model's



(a)

capacity for detection. The Faster RCNN with ResNet50 model has been proposed an 89.47 mAP score and a 4.312 average detection speed per second. The detection of two classes; flower and bud represent the flower growing stages that can be helpful to decide the harvesting time for the Marigold flowers. This research provides a pathway for the researchers who are working for the automatic detection and harvesting of flowers other than the Marigold.

REFERENCES

- [1] M. A. Wani et al., "Floriculture sustainability initiative: The dawn of New Era," in *Sustainable Agriculture Reviews 27*, Cham: Springer International Publishing, 2018, pp. 91–127.
- [2] Jagdish, "Marigold Cultivation Income, yield, project report," Agri Farming, 08-Sep-2019. [Online]. Available: <https://www.agrifarming.in/marigold-cultivation-income-yield-project-report/>. [Accessed: 20-Feb-2023].
- [3] "Harvesting and handling cut flowers," Center for Agriculture, Food, and the Environment, 04-Aug-2016. [Online]. Available: <https://ag.umass.edu/greenhouse-floriculture/fact-sheets/harvesting-handling-cut-flowers>. [Accessed: 20-Feb-2023].
- [4] Y. Jiang, C. Li, R. Xu, S. Sun, J. S. Robertson, and A. H. Paterson, "DeepFlower: A deep learning-based approach to characterize flowering patterns of cotton plants in the field," *Plant Methods*, vol. 16, no. 1, p. 156, 2020.
- [5] D. Thi Phuong Chung and D. Van Tai, "A fruits recognition system based on a modern deep learning technique," *J. Phys. Conf. Ser.*, vol. 1327, no. 1, p. 012050, 2019.
- [6] Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Comput. Electron. Agric.*, vol. 70, no. 1, pp. 96–104, 2010.
- [7] Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "DeepFruits: A fruit detection system using deep neural networks," *Sensors (Basel)*, vol. 16, no. 8, p. 1222, 2016.
- [8] T. Abbas et al., "Deep neural networks for automatic flower species localization and recognition," *Comput. Intell. Neurosci.*, vol. 2022, p. 9359353, 2022.
- [9] N. Kumar et al., "Leafsnap: A computer vision system for automatic plant species identification," in *Computer Vision – ECCV 2012*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 502–516.
- [10] Joly et al., "Interactive plant identification based on social image data," *Ecol. Inform.*, vol. 23, pp. 22–34, 2014.
- [11] Patel and S. Patel, "An Optimized Deep Learning Model for Flower Classification Using NAS-FPN and Faster R-CNN," *International Journal of Scientific & Technology Research*, vol. 9, no. 03, pp. 5308–5318, 2020.
- [12] D. Wu et al., "Detection of Camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation," *Appl. Sci. (Basel)*, vol. 12, no. 22, p. 11318, 2022.
- [13] S. Nuanmeesri, S. Chopvitayakun, P. Kadmatekarun, and L. Poomhira, "Marigold flower disease prediction through deep neural network with multimodal image," *Int. J. Eng. Trends Technol.*, vol. 69, no. 7, pp. 174–180, 2021.
- [14] "LabelImg," PyPI. [Online]. Available: <https://pypi.org/project/labelImg/1.4.0/>. [Accessed: 20-Feb-2023].
- [15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, 2019.
- [16] T. Luke and N. Geoff, *Improving deep learning using generic data augmentation*. 2017.
- [17] Z. Keita, "Five simple image data augmentation techniques to mitigate overfitting in Computer Vision," *Towards Data Science*, 19-Mar-2021. [Online]. Available: <https://towardsdatascience.com/simple-image-data-augmentation-technics-to-mitigate-overfitting-in-computer-vision-2a6966f51af4>. [Accessed: 20-Feb-2023].
- [18] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sens. (Basel)*, vol. 13, no. 1, p. 89, 2020.
- [19] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, "An evaluation of deep learning methods for small object detection," *J. Electr. Comput. Eng.*, vol. 2020, pp. 1–18, 2020.
- [20] I. Patel and S. Patel, "A comparative analysis of applying object detection models with transfer learning for flower species detection and classification," *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 303–312, 2020.
- [21] Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, 2022.
- [22] Y. Gu, S. Wang, Y. Yan, S. Tang, and S. Zhao, "Identification and analysis of emergency behavior of cage-reared laying ducks based on YoloV5," *Agriculture*, vol. 12, no. 4, p. 485, 2022.
- [23] G. B. Rajendran, U. M. Kumarasamy, C. Zarro, P. B. Divakarachari, and S. L. Ullo, "Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM classifier on hybrid pre-processing remote-sensing images," *Remote Sens. (Basel)*, vol. 12, no. 24, p. 4135, 2020.
- [24] R. Potter, "What is the use and purpose of image annotation in object detection?," *Becoming Human: Artificial Intelligence Magazine*, 19-May-2021. [Online]. Available: <https://becominghuman.ai/what-is-the-use-and-purpose-of-image-annotation-in-object-detection-8b7873a14cd0>. [Accessed: 20-Feb-2023].
- [25] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Comput. Electron. Agric.*, vol. 178, no. 105760, p. 105760, 2020.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv [cs.CV]*, 2015.
- [27] M. Shen et al., "Multi defect detection and analysis of electron microscopy images with deep learning," *Comput. Mater. Sci.*, vol. 199, no. 110576, p. 110576, 2021.
- [28] S. Aparna, K. Muppavaram, C. C. V. Ramayanam, and K. S. S. Ramani, "Mask RCNN with RESNET50 for dental filling detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 10, 2021.
- [29] S. Patel, R. Patel, N. Ganatra, and A. Patel, "Spatial feature fusion for biomedical image classification based on ensemble deep CNN and transfer learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, 2022.
- [30] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, "Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches," *arXiv [cs.CV]*, 2022.
- [31] G. Boesch, "Deep Residual networks (ResNet, ResNet50) - 2023 guide," *viso.ai*, 01-Jan-2023. [Online]. Available: <https://viso.ai/deep-learning/resnet-residual-neural-network/>. [Accessed: 20-Feb-2023].
- [32] He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv [cs.CV]*, 2015.
- [33] E. Koech, "Cross-entropy loss function," *Towards Data Science*, 02-Oct-2020. [Online]. Available: <https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>. [Accessed: 20-Feb-2023].
- [34] S. Reshma Prakash and P. Nath Singh, "Object detection through region proposal based techniques," *Mater. Today*, vol. 46, pp. 3997–4002, 2021.
- [35] Rosebrock, "Intersection over Union (IoU) for object detection," *PyImageSearch*, 07-Nov-2016. [Online]. Available: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. [Accessed: 20-Feb-2023].
- [36] F. Gad, "Faster R-CNN explained for object detection tasks," *Paperspace Blog*, 13-Nov-2020. [Online]. Available: <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>. [Accessed: 20-Feb-2023].
- [37] S. Ananth, "Faster R-CNN for object detection," *Towards Data Science*, 09-Aug-2019. [Online]. Available: <https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>. [Accessed: 20-Feb-2023].
- [38] H. Yan, C. Chen, G. Jin, J. Zhang, X. Wang, and D. Zhu, "Implementation of a modified faster R-CNN for target detection

- technology of coastal defense radar,” *Remote Sens. (Basel)*, vol. 13, no. 9, p. 1703, 2021.
- [39] Du, R. Zhang, and X. Wang, “Overview of two-stage object detection algorithms,” *J. Phys. Conf. Ser.*, vol. 1544, no. 1, p. 012033, 2020.
- [40] T. Hoerer and C. Kuenzer, “Object detection and image segmentation with deep learning on Earth observation data: A review-part I: Evolution and recent trends,” *Remote Sens. (Basel)*, vol. 12, no. 10, p. 1667, 2020.
- [41] Naukri.com. [Online]. Available: <https://www.naukri.com/learning/articles/object-detection-using-rcnn/>. [Accessed: 20-Feb-2023].
- [42] Researchgate.net. [Online]. Available: https://www.researchgate.net/profile/Hadi-Ghahremanzhad/publication/346061113_Vehicle_Classification_in_Video_Using_Deep_Learning/links/5fb985ef92851c933f4d56be/Vehicle-Classification-in-Video-Using-Deep-Learning.pdf. [Accessed: 20-Feb-2023].
- [43] “No title,” Modelzoo.co. [Online]. Available: <https://modelzoo.co/model/objectdetection>. [Accessed: 24-Feb-2023].
- [44] S. Patel, “Diabetic Retinopathy Detection and Classification using Pre-trained Convolutional Neural Networks,” *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 1082–1087, 2020.
- [45] T. Agrawal, *Hyperparameter optimization in machine learning: Make your machine learning and deep learning models more efficient*. Berkeley, CA: Apress, 2021.
- [46] Kumar, R. C. Joshi, M. K. Dutta, M. Jonak, and R. Burget, “Fruit-CNN: An efficient deep learning-based fruit classification and quality assessment for precision agriculture,” in *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2021, pp. 60–65.
- [47] S. Cheng, K. Zhao, and D. Zhang, “Abnormal water quality monitoring based on visual sensing of three-dimensional motion behavior of fish,” *Symmetry (Basel)*, vol. 11, no. 9, p. 1179, 2019.
- [48] J. Huang et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [49] International Journal of Computer (IJC), “View of marigold blooming maturity levels classification using machine learning algorithms,” *ijcjournal.org*. [Online]. Available: <https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/1870/686>. [Accessed: 23-Mar-2023].
- [50] S. Bondre and U. Yadav, “Automated flower species identification by using deep convolution neural network,” in *Intelligent Data Engineering and Analytics*, Singapore: Springer Nature Singapore, 2022, pp. 1–10.
- [51] T.-H. Hsu, C.-H. Lee, and L.-H. Chen, “An interactive flower image recognition system,” *Multimed. Tools Appl.*, vol. 53, no. 1, pp. 53–73, 2011.
- [52] A. Rawat and R. Kaur, “Proposed methodology of supervised learning technique of flower feature recognition through machine learning,” *Jetir.org*. [Online]. Available: <https://www.jetir.org/papers/JETIR1907J60.pdf>. [Accessed: 23-Mar-2023].