



Mango Orchard Aerial Image Dataset for Research on Fruit Trees with UAVs in the Precision Agriculture

Kavita Houde¹, Parshuram Kamble¹, Ravindra Hegadi¹

10.18805/ag.D-6385

ABSTRACT

Background: Agriculture plays a vital role in the economic development of many countries worldwide. The integration of cost-effective artificial intelligence (AI) solutions into farming practices has the potential to significantly enhance productivity and improve the livelihood of farmers. One of the major challenges in automating agricultural tasks such as pesticide spraying on fruit trees, is the lack of sufficient labeled datasets. This limitation hinders the development of robust AI models for accurate tree detection, segmentation and spraying. A well-structured and diverse database is therefore essential for training effective AI models that can support precision agriculture.

Methods: In this work, we propose the systematic collection of aerial images of mango orchards and the creation of mango fruit image database captured using a DJI Air 2S UAV (Unmanned Aerial Vehicle). We also proposed image processing algorithms to divide, classify and annotate the images. These images are collected from different Indian farms where high numbers of mango orchards are found. The database contains 3917 mango tree images, which were manually sorted after collection. The images, which are large in size, are divided into smaller frames to make further image processing tasks easy and then those frames are classified based on the tree availability using different image processing techniques such as HSV, tree contours, contour area and eccentricity threshold. The count we got is 69607 frames with the presence of trees and 8713 frames with the absence of trees. Later, image processing algorithm was designed to annotate tree positions in images in (You Only Look Once) YOLO format within a specified directory structure, with tasks such as identifying tree positions based on color thresholds and drawing bounding boxes around them.

Results: A total of 58,608 frames were annotated using the YOLO format. Frames that did not contain mango trees were excluded. Annotation was performed through user-interaction-based methods to ensure high-quality labeling. As a result, all annotated images are confirmed to contain mango trees or parts of mango trees only. These images are accessible to anyone interested in the research in related domains. Through this paper, we are publishing our UAV-captured original mango orchard image dataset and its auto-annotated image dataset for the Image processing, AI, ML and DL based UAVs application developments.

Key words: Aerial images, Image database, Mango orchards, Tree dataset, UAVs.

INTRODUCTION

The advancement of precision agriculture heavily relies on the integration of AI and computer vision techniques to automate innovation farming tasks. The advent of UAVs, widely known as drones, has become an invaluable tool in various fields due to their ability to provide high-resolution, cost-effective and real-time data (Goodchild, 2007). UAVs have brought about a paradigm shift in agricultural practices, particularly in the domain of orchard management and fruit tree cultivation (Kumar *et al.*, 2021). Their flexibility, rapid deployment and manoeuvrability make them invaluable in various applications, including environmental monitoring, precision agriculture, infrastructure inspection, mapping, security and surveillance (Tuia *et al.*, 2016). The significance of UAVs in image collection lies in their ability to provide high-resolution, real-time, cost-effective data and improved accessibility to challenging terrains, making them versatile tools across industries such as agriculture, disaster response and entertainment for a wide range of applications (Kumar *et al.*, 2021). The author Girijalaxmi *et al.* (2024) developed an algorithm to calculate the distance between trees for agricultural applications. The integration of UAV technology in mango tree data collection

¹Department of Computer Science, Central University of Karnataka, Kadaganchi, Kalaburagi-585 367, Karnataka, India.

Corresponding Author: Ravindra Hegadi, Department of Computer Science, Central University of Karnataka, Kadaganchi, Kalaburagi-585 367, Karnataka, India.

Email: rshegadi@gmail.com

ORCIDs: 0009-0002-2262-8002, 0000-0003-4401-575X, 0000-0002-5176-6490

How to cite this article: Houde, K., Kamble, P. and Hegadi, R. (2025). Mango Orchard Aerial Image Dataset for Research on Fruit Trees with UAVs in the Precision Agriculture. *Agricultural Science Digest*. 1-11. doi: 10.18805/ag.D-6385.

Submitted: 06-06-2025 **Accepted:** 16-12-2025 **Online:** 06-01-2026

stands as a testament to this transformation, ushering in an era of precision agriculture for one of the world's most beloved fruits (Ma *et al.*, 2019).

The author Houde *et al.* (2024) used the online mango tree videos to generate images for the development of a tree detection model. A robust UAVs based images dataset is crucial for the development of UAV and deep learning-based AI applications (Ma *et al.*, 2019). However, a major

bottleneck in developing robust AI models for tree-level applications such as spraying, yield estimation and health monitoring is the lack of publicly available, high-resolution and well-annotated datasets, especially for fruit trees like mangoes, apple, citrus *etc.* Existing agricultural datasets predominantly focus on common field crops and do not offer object-level annotations or imagery tailored to orchard environments. Even when such datasets exist, they are typically not captured using UAVs, making them unsuitable for aerial tree-level analysis. UAVs can generate large volumes of high-resolution images during a single flight. A robust database is essential to handle massive data efficiently and ensure quick and reliable retrieval (Zhu *et al.*, 2017). A well-designed database structure helps in organizing UAV-generated images systematically, making it easier to access specific datasets, locations, or time frames (Sawat *et al.*, 2016). This organization enhances data accessibility for analysis, research and decision-making, facilitating seamless integration with deep learning frameworks supporting tasks such as image annotation, training and validation (Zhang *et al.*, 2016). Additionally, it allows for the exploration of temporal trends, enhances data preprocessing for machine learning models and ensures the reliability and scalability necessary for the advancement of AI applications in fields like object detection, segmentation and environmental monitoring. UAVs often carry various sensors capturing different types of data, such as visual, infrared, or multispectral imagery (Jose *et al.*, 2021). Since UAVs can capture images at different time intervals, a database supporting temporal analysis allows researchers and analysts to track changes over time (Jr *et al.*, 2013). This is valuable for monitoring dynamic environments, such as urban development or natural resource changes.

To address this gap, we present a comprehensive UAV-based dataset of mango trees with YOLO-format annotations, specifically designed to support detection and segmentation tasks in real-world orchard settings.

UAVs, often known as drones, have significantly transformed various industries by providing advanced, high-resolution imaging capabilities and unmatched flexibility in operations. For our data collection, we used the UAV DJI Air 2S (Fig 1).

Literature review

The recent surge in studies exploring the use of unmanned aerial vehicles (UAVs) in various fields highlights their versatility and capability in data collection and analysis. Khan *et al.* (2017) illustrate UAVs' transformative role in urban planning and traffic management, providing a bird's-eye view that surpasses traditional ground-based methods. UAVs equipped with high-resolution cameras can capture detailed imagery of urban traffic, processed using advanced computer vision algorithms for real-time monitoring and object detection.

UAVs were also used for remote sensing and various civil applications. Shakhathreh *et al.* (2019) and Mithra *et al.* (2021) provided a comprehensive overview of UAV uses, from image and video data collection to environmental monitoring and infrastructure assessment, addressing



Fig 1: UAV DJI Air 2S used for data collection.



Fig 2: Drone navigation pattern view of the selection area in the dronelink application.

current challenges and future research directions. These studies reflect the expanding role of UAVs in various sectors, from urban development to agriculture and their intersection with advanced technologies like AI and machine learning. Integrating these tools and techniques paves the way for innovative solutions and enhanced efficiency in data collection, analysis and application across multiple domains.

Data collection overview

For the proposed work, the data was collected from the Bagalkot, Hubli, Belagavi and Dharwad districts of

Karnataka state in the southern part of India and Dapoli, Ratnagiri district of Maharashtra state. The selection of these regions for UAV data collection is based on several factors, primarily the abundance of mango trees available in these regions Vinita *et al.* (2022) and Ausari *et al.* (2023). These regions experience a subtropical climate, which is conducive to mango cultivation. The warm temperatures and moderate rainfall during the monsoon season provide an ideal environment for mango trees to thrive and bear fruit Ray *et al.* (2022) and Gulati *et al.* (2021). These regions boast a rich agricultural landscape with extensive mango orchards and plantations. The fertile soil and availability of

Table 1: Details of images collected through the drone.

Date	Time	Altitude (Meter)	Front overlap (%)	Side overlap (%)	Speed (Km/h)	Ground sample distance	Images collected
14/01/24	3:45 pm-4:30 pm	50	75	70	10	1.42 cm/px	169
15/01/24	9:30 am-10 am	30	25	25	25	0.85 cm/px	10
15/01/24	10 am-10:30 am	30	40	40	20	0.85 cm/px	131
15/01/24	11:10 am-11:20 am	30	40	40	20	1.42 cm/px	33
15/01/24	11:27 am-11:31 am	40	40	40	20	1.14 cm/px	27
15/01/24	11:41 am-11:46 am	30	30	30	20	0.85 cm/px	27
15/01/24	12:51 pm-12:56 pm	40	50	50	20	1.42 cm/px	52
15/01/24	2:51 pm-2:56 pm	30	40	40	30	0.85 cm/px	75
15/01/24	3:29 pm-3:56 pm	40	50	50	25	1.14 cm/px	271
15/01/24	4:21 pm-4:24 pm	40	40	40	25	1.14 cm/px	26
20/01/24	10:17 am-10:20 am	40	50	50	25	1.14 cm/px	24
20/01/24	11:02 am-11:05 am	40	60	60	25	1.14 cm/px	24
20/01/24	11:23 am-12:00 pm	45	50	50	25	1.14 cm/px	28
20/01/24	12:29 pm-12:32 pm	40	70	70	25	1.14 cm/px	38
20/01/24	12:58 pm-1:09 pm	40	40	40	25	1.14 cm/px	132
20/01/24	1:28 pm-1:38 pm	40	40	40	25	1.14 cm/px	68
20/01/24	3:12 pm-3:16 pm	40	60	60	25	1.14 cm/px	24
20/01/24	3:31 pm-3:35 pm	40	50	50	32.2	1.14 cm/px	29
20/01/24	4:28 pm-4:33 pm	40	50	50	30	1.14 cm/px	60
20/01/24	4:46 pm-4:49 pm	50	70	70	25	1.14 cm/px	39
20/01/24	5:36 pm-5:43 pm	50	60	60	25	1.14 cm/px	101
20/01/24	5:52 pm-5:57 pm	35	50	50	30	1.14 cm/px	21
21/01/24	9:50 am-10:01 am	50	60	60	25	1.42 cm/px	131
21/01/24	10:40 am-10:53 am	40	60	60	30	1.14 cm/px	250
21/01/24	11:26 am-11:49 am	40	60	60	30	1.14 cm/px	257
21/01/24	11:58 am-12:01 pm	40	60	60	30	1.14 cm/px	43
26/01/24	12:56 pm-1:10 pm	60	70	70	25	1.70 cm/px	232
26/01/24	1:26 pm-1:44 pm	60	80	80	25	1.70 cm/px	372
26/01/24	2:02 pm-2:24 pm	60	80	80	25	1.70 cm/px	423
26/01/24	04:10 pm-04:26 pm	60	80	80	25	1.70 cm/px	180
26/01/24	4:51 pm-5:00 pm	60	80	80	32.2	1.70 cm/px	81
26/01/24	5:56 pm - 6:07 pm	60	70	70	25	1.70 cm/px	177
23/03/24	3:26 pm to 3:40 pm	50	50	50	40	1.42 cm/px	67
23/03/24	3:57 pm to 4:30 pm	50	50	50	35	1.42 cm/px	195
23/03/24	4:50 pm to 5:00 pm	50	50	50	35	1.42 cm/px	42
24/03/24	8:50 am to 8:30 am	50	50	50	35	1.42 cm/px	97
Total images							3917

water resources support the cultivation of mango trees on a large scale.

Hardware and software used for data collection

Dronelink android-based mapping software is used over a Redmi Note 11S cellphone for drone navigation, control and image capturing. DroneLink is a comprehensive software platform designed to streamline and enhance the process of aerial mapping and data collection using drones. It offers intuitive mission planning tools, real-time monitoring and advanced data processing capabilities. Fig 2 shows the map named “Dharwad2” captured at an altitude of 30 mts. The total covered area is 7.5 hectares, with a front overlap of 40% and a side overlap of 40%. In DroneLink, the “normal” pattern refers to a predefined flight path or trajectory designed to cover a specific area during aerial mapping missions. This pattern typically involves flying the drone in a systematic grid-like or zigzag pattern over the designated area, ensuring comprehensive coverage and consistent data collection. The parameters “front overlap 40% and “side overlap 40% specify the amount of overlap between consecutive images captured by the drone. Table 1 shows some common parameters used for all the images captured through the drone.

The dataset collected through the UAV comprises 3917 RGB images with dimensions of 5472×3648 pixels each, featuring a standard resolution of 72 dpi. The gimbal pitch was set to -90 degrees, capturing images with a downward perspective. The lens used had a focal length of 8.5mm, mounted on a sensor measuring $13.20 \text{ mm} \times 8.80 \text{ mm}$. The images were captured over a period spanning from January 14th to March 24th, 2024. These images occupy a disk space ranging from 10 MB to 14 MB per image, reflecting the high-resolution nature of the dataset. These images offer a comprehensive visual dataset for analysis and research. The consistent normal pattern of data collection and high-resolution imagery make this dataset valuable for deep learning-based AI applications and further study in fields like object detection, segmentation and landscape analysis.

Geographic locations

The UAV-collected dataset encompasses the rural and urban regions of Bagalkot, Hubli-Dharwad and Belagavi in Karnataka state and Dapoli, Ratnagiri in Maharashtra state of India. The geographical coordinates range from latitude X to Y and longitude A to B. Google Play Store GPS Map Camera, an Android-based open source application, is used to capture Latitude and Longitude values from

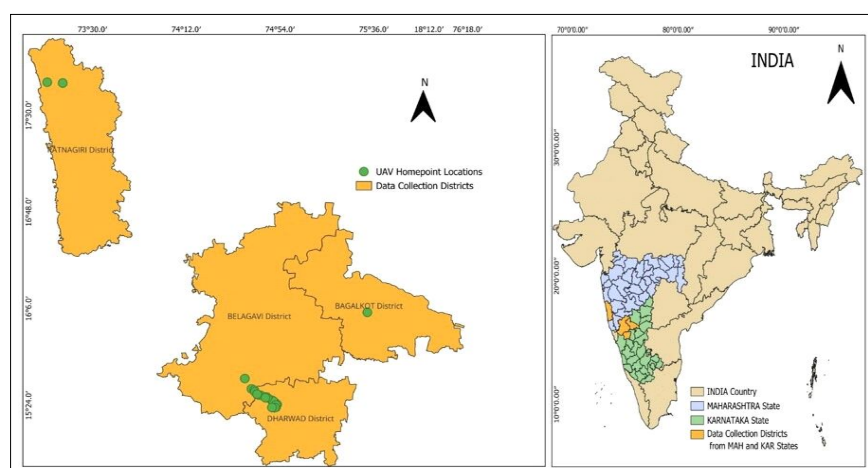


Fig 3: Map of the region under study.



Fig 4: Original image captured by UAV.

the UAV Home-point locations which provides location information.

The maps depicting the data-collected area were crafted using QGIS, a powerful and versatile GIS software. In Fig 3, the green color dots from the Maharashtra and Karnataka districts illustrate the data-collected area, generated using QGIS and the spatial patterns of the observed phenomena. The color-coded layers in the map showcase variations in terrain characteristics, aiding in the interpretation of field observations. QGIS's capabilities in geospatial analysis played a pivotal role in uncovering patterns and trends within the study area.

The maps depicting the data-collected area were crafted using QGIS, a powerful and versatile GIS software. QGIS allows for seamless integration of various data layers, including aerial imagery and geospatial features. The created maps provide a comprehensive visualization of the study area, highlighting specific points of interest and the spatial distribution of collected data.

MATERIALS AND METHODS

The proposed research work is carried out at the Central University of Karnataka, Kalaburagi, India during the year 2023-24. The UAV-captured image processing was conducted using a laptop with 32 GB of RAM and Windows 10 OS Multiprocessor Free. The analysis and processing workflows were seamlessly executed within a Jupyter Notebook environment, offering an interactive and flexible platform for code development, data visualization and comprehensive analysis. This integrated setup, combining GPU acceleration with the versatile capabilities of Jupyter Notebook, facilitated efficient and thorough analysis of the UAV-captured images. The Python version used for this processing was 3.9.13, providing the necessary programming environment for conducting advanced image analysis tasks.

Image preprocessing

UAV-collected images from the different locations of the mango orchard fields are stored under one main base directory with the name Tree Dataset with different subdirectories followed by the names field1, field2,..., field36. Images captured in each field are stored under a single subdirectory. The sample of the original captured image is shown in Fig 4.

These Images often have large dimensions that may need to be processed in smaller segments. A mathematical approach is proposed to efficiently divide a high-dimensional image into smaller frames without losing crucial information. When working with large images, breaking them into smaller frames is essential for efficient processing, analysis, or further applications. The primary objective is to divide the original image into frames of equal size (1024 × 1024 pixels) while incorporating a specified overlap to facilitate seamless stitching and comprehensive coverage. This is achieved through a systematic approach that calculates the overlap in the directions of the x and y axes based on the original image dimensions and the

desired frame size. The overall process is described with the Number of Frames Calculation, Overlap Calculation and Slicing Operation.

Number of frames calculation

The number of frames in the x and y directions is determined using the formulas

$$\text{num_frames_x} = W/F$$

$$\text{num_frames_y} = H/F$$

The equations for calculating the overlap in the x and y directions are based on the original image width W , height H and the desired frame size F to extract frames (sliced image) from the original image. These formulas ensure that all areas of the original image are covered by frames without any gaps.

Overlap calculation

The overlap in the x and y directions is calculated as follows:

$$O_x = \frac{(W - \text{num_frames_x} \times F)}{W - \text{num_frames_x} \times F}$$

$$O_y = \frac{(H - \text{num_frames_y} \times F)}{H - \text{num_frames_y} \times F}$$

Where,

O_x = Overlap in the x - direction (horizontal overlap).

O_y = Overlap in the y - direction (vertical overlap).

These equations determine the overlap between adjacent frames to cover the entire image area without gaps.

Slicing operation

In the image slicing process, the nested loops iterate over the image, creating frames of the specified size with the desired overlap. Each frame is created using the slicing operation and frames are appended to a list and then the list of frames is returned. This process divides the image into frames using the calculated overlap values to ensure uniform frame extraction. The mathematical expression to perform the slicing operation is presented with the nested loops:

For y in the range $(0, H - F + 1, F - O_y)$

For x in the range $(0, W - F + 1, F - O_x)$

Frame = Image [$y : y + F, x : x + F$]

Here, the starting index for height and width is determined by the loop variables y and x and the frame size is F . The overlap O is subtracted to ensure the frames have the desired spacing. The frames are then collected and put into a list.

The dimension of the original UAV captured images (e.g. DJI 0172.jpg) is 5472 × 3648 pixels. For the image-slicing process, we considered frame size $F = 1024$ and the calculated Overlap in x - direction is 70.40 pixels and in y - direction is 192.00 pixels. Through this process, 20 frames are generated from each UAV image. This overall process took 61 minutes for 3917 original images. Finally, we obtained 78,340 frames among all the UAV-captured original tree images.

The process begins by representing the image through its essential dimensions - height, width and channels. Subsequently, a systematic slicing process is implemented using nested loops, allowing the creation of frames with specified dimensions and desired overlap. Each frame is generated through slicing operations and these frames are then collected and appended to a list. Ultimately, the outcome is a list containing the smaller frames, providing a manageable and organized representation of the original image. Through the above process, the frames are generated with extended names of the original image name (e.g. DJI_0172_frame_1.jpg) and stored in a folder created with the name of the image. These frames are 24-bit RGB images with dimensions of 1024×1024 pixels. The size of the frames may vary depending on the content of these frames. A Sample frame is shown in Fig 5.

After completing the slicing process, two Excel sheets are generated and stored in the directories: "*Tree_Dataset\Field1\DJI_0172\DJI_0172_frame_details.xlsx*" and "*Tree_Dataset\Field1\All_original_image_details.xlsx*".

These Excel sheets contain essential details and metadata extracted during the image processing and slicing. The first Excel sheet contains Frame Number, Frame Filename, Frame Width, Frame Height, Image Mode, Frame Bit Depth, Frame Timestamp and File Size (in KB), while the second file contains Name, Width, Height, Size (in MB), Image Type, Date, Latitude, Longitude, Number of Frames, Time to Divide Frames (in sec.) and Time to Save Frames (in sec.). These Excel sheets provide comprehensive information about the sliced frames and original images, facilitating further analysis and research.

Classification of divided frames

Post-framing operation, there is every possibility that some frames may not contain the trees as part of the images. Such images shall be discarded from the image database through the process described here. The divided frames that are already stored in the separate folders (e.g., for image, DJI_0172.jpg a folder will be created with the name DJI_0172) with a subfolder containing all the frames of respective images need to be classified through the automated process. Algorithm 1 performs the Automated Classification of images (divided frames) to find the availability of the tree in the frames.

Algorithm 1: Automated Classification of images (divided frames) to find the availability of the tree.

1. Initialization

Input: Directory containing image files.

Output: Images categorized as 'Tree Present' or 'Tree Absent'.

2. Image preprocessing

- Load images from the specified directory.
- For each image, check and resize to 640×640 pixels if necessary.

3. Color space conversion

- Convert the image from BGR to HSV color space.

4. Green color identification

- Define lower and upper bounds for the green color in HSV.
- Create a binary mask to isolate regions with green color.

5. Contour detection

- Detect contours in the binary mask.
- Identify the largest contour, if any are found.

6. Feature analysis

For the largest contour:

- Calculate the area of the contour.
- Fit an ellipse to the contour.
- Calculate the eccentricity of the fitted ellipse.

7. Decision making

Based on predefined thresholds for area and eccentricity, determine if a tree is present or absent in the image.

8. Image categorization and storage

- Save images in respective folders ("Tree Present" or "Tree Absent").
- Store the EXIF data of the original image in the processed image.

After completing the automated classification process, three Excel sheets are generated and stored in the directories: "*Tree_Dataset\Field1\DJI_0172\ Frames_TP_TA_*"



Fig 5: Sample frame generated through slicing process.

Table 2: Tree classification sample details for frames of a single original image.

Frame filename	Frame shape	Tree present
DJI_0172_frame_1	(1024, 1024, 3)	TRUE
DJI_0172_frame_2	(1024, 1024, 3)	TRUE
DJI_0172_frame_3	(1024, 1024, 3)	TRUE
DJI_0172_frame_4	(1024, 1024, 3)	TRUE
DJI_0172_frame_5	(1024, 1024, 3)	TRUE

Details.xlsx", "Tree_Dataset\Field1\ Images_TP_TA_Details.xlsx" and "Tree_Dataset\Fields_TP_TA_Details.xlsx". The first Excel sheet is generated for the details, indicating a Boolean value for the tree availability in the divided frames for each divided frame in the same location of the folder. Tree present frames are stored in the "Tree Present" folder and tree absent frames are stored under the "Tree Absent" folder under the same directory. The sample data is shown in Table 2.

The second column in Table 3 shows the number of frames containing the trees and the third column shows the number of frames not containing the trees after the images are segmented into frames.

In our analysis of the Tree Dataset, we delved into the presence and absence of trees across different fields. The findings revealed interesting variations: Field1 showed a substantial presence of 59 trees with only one absence recorded. Finally, we obtained a total of 69607 frames with trees present and 8713 frames with trees absent among all fields containing tree images captured by UAV. The field-wise frames with the presence or absence of trees are shown in Table 4. The process of classifying the frames took 56 minutes.

RESULTS AND DISCUSSION

Annotations and labeling

Automatically annotating images, also known as automated or algorithmic annotation, offers several advantages, particularly in terms of efficiency, scalability and reducing human labor. Automated annotation processes are significantly faster than manual annotation and algorithms can process and annotate large datasets in a fraction of the time. In this work, we adopted automation of annotations in the YOLO format, a popular format for object detection tasks. To achieve this, the Python script is used, which leverages the OpenCV library to detect and annotate trees from the divided and classified frames. The script processes images to identify tree-like structures and generates annotations in the YOLO format, enabling compatibility with YOLO-based object detection models. The Algorithm 2 shows the steps involved in the proposed automatic annotation of trees from the frames.

Algorithm 2: Automatic annotation of trees from frames with user confirmation.

Input: Image path.

Output: Bounding Box Annotated image, YOLO format annotations text file.

Reads an image \$I\$ from a specified path.

Convert the image to HSV color space.

Specify lower bound and upper bound color thresholds in HSV Space.

Create a binary mask for green regions.

Find contours in the binary mask.

Iterate through all contours.

Get bounding box coordinates of the contour and

Calculate \$bounding_rect_area\$
if \$bounding_rect_area > 5000\$:

Begin

Draw the bounding rectangle on the image

Display the image with the drawn bounding box to the user.

Ask the user bounding box to the annotations notebook (yes/no).

If the user answers "yes":

Normalize bounding box coordinates values between 0 and 1

Save YOLO format annotations to a text file.

Write the DataFrame to an Excel file for each subdirectory,

\\summarizing annotation statistics.

If the user chooses "no":

Proceed to the next bounding box.

End

Store global counts in a separate Excel file for comprehensive analysis.

One of the critical aspects of image processing in computer vision is accurately annotating objects within images. We used an interactive approach to bounding box annotation which offers a streamlined method for annotators to mark regions of interest. This process not only enhances annotation accuracy but also improves the efficiency of data labeling tasks. In a typical interactive bounding box annotation workflow, annotators are presented with images containing objects of interest. Using an annotation tool or software, annotators can draw bounding boxes around these objects. It is further enhanced for incorporating user interaction, which can provide valuable control over annotation decisions.

An image processing pipeline detects multiple objects within an image. Rather than automatically adding all detected bounding boxes to the annotation file, an interactive approach prompts annotators to review each detected object individually. Annotators are presented with each bounding box overlaid on the image and are asked whether to include or exclude that specific object from the annotation.

This interactive decision-making process empowers annotators to make informed judgments based on their expertise and domain knowledge. It allows annotators to verify the accuracy of detection and effectively exclude false positives or irrelevant objects. Additionally, annotators can prioritize objects of interest, ensuring that crucial elements

Table 3: Sample tree classification details with respect to original images.

Filename	Total tree present	Total tree absent
DJI_0164.jpg	19	1
DJI_0165.jpg	18	2
DJI_0166.jpg	16	4
DJI_0167.jpg	13	7
DJI_0168.jpg	14	6
DJI_0169.jpg	16	4

are accurately annotated while reducing annotation noise. Integrating user interaction in bounding box annotation enhances annotation quality and contributes to creating more robust and reliable datasets for training machine learning models. By incorporating annotator feedback and expertise into the annotation process, interactive bounding box annotation becomes a valuable tool in developing accurate and high-quality datasets for computer vision applications.

A binary mask M is created to isolate green regions in the image using Eqn. 1.

$$M(x,y) = \begin{cases} 1 & \text{if } H_{\min} \leq H(x,y) \leq H_{\max}, S_{\min} \leq S(x,y) \leq S_{\max}, \\ & V_{\min} \leq V(x,y) \leq V_{\max} \\ 0 & \text{otherwise} \end{cases} \quad \dots(1)$$

$M(x, y)$ = value of the binary mask at pixel (x, y) .

This expresses that the binary mask value at pixel (x, y) is 1 if the pixel's HSV values are within the specified range and 0 otherwise. Fine-tuning these bounds ensures that the binary mask effectively highlights the green regions in the image as shown in Fig 6. Fig 7(a) shows the result of applying the proposed contour detection method over the binarized image shown in Fig 6. The significance of the bounding box filtering lies in its capacity to streamline computational efforts, focusing attention on meaningful regions and contributing to the overall effectiveness of the computer vision application. One such frame in which the object (tree region) is extracted by applying a bounding box using the proposed method is shown in Fig 7(b).

The bounding box is added to the image one by one and asks the user whether to add the current bounding box in the annotation file, which is highlighted using a red border as shown in Fig 8. When the user answers "yes" to add a bounding box to the YOLO format annotations, the algorithm normalizes the bounding box coordinates to values between 0 and 1.

The dimensions of the object (tree region) are normalized to obtain coordinate values between 0 and 1. We first extracted the Bounding box coordinates to normalize the (x, y, w, h) values. Here the image dimensions are $image_{height}$ (height) and $image_{width}$ (width). The following values are computed:

Normalized x-coordinate of the bounding box center:

$$\frac{x + \frac{w}{2}}{image_{width}}$$

Normalized y-coordinate of the bounding box center:

$$\frac{y + \frac{h}{2}}{image_{height}}$$

Normalized width of the bounding box: $\frac{w}{image_{width}}$

Normalized height of the bounding box: $\frac{h}{image_{height}}$

Where $x + \frac{w}{2}$ calculates x-coordinate of the center of the bounding box and $y + \frac{h}{2}$ calculates y-coordinate of the center of the bounding box. Dividing the x and y coordinate values by $image_{width}$ and $image_{height}$ respectively normalizes these coordinates to the range $[0, 1]$. The rounding to 6 decimal places is applied for precision. The resulting normalized coordinates fall within the range $[0, 1]$, making

Table 4: All fields divided frames tree classification details.

Field directory	Total tree present	Total tree absent
Tree_Dataset\Field1	3025	335
Tree_Dataset\Field2	183	17
Tree_Dataset\Field3	1683	157
Tree_Dataset\Field4	626	34
Tree_Dataset\Field5	527	13
Tree_Dataset\Field6	507	33
Tree_Dataset\Field7	971	69
Tree_Dataset\Field8	1262	238
Tree_Dataset\Field9	4956	464
Tree_Dataset\Field10	482	38
Tree_Dataset\Field11	381	99
Tree_Dataset\Field12	370	110
Tree_Dataset\Field13	529	31
Tree_Dataset\Field14	704	56
Tree_Dataset\Field15	2288	352
Tree_Dataset\Field16	1192	168
Tree_Dataset\Field17	350	130
Tree_Dataset\Field18	509	71
Tree_Dataset\Field19	978	222
Tree_Dataset\Field20	696	84
Tree_Dataset\Field21	1877	143
Tree_Dataset\Field22	381	39
Tree_Dataset\Field23	2143	477
Tree_Dataset\Field24	4341	659
Tree_Dataset\Field25	3401	1739
Tree_Dataset\Field26	755	105
Tree_Dataset\Field27	4280	360
Tree_Dataset\Field28	6723	717
Tree_Dataset\Field29	8073	387
Tree_Dataset\Field30	3155	445
Tree_Dataset\Field31	1404	216
Tree_Dataset\Field32	3298	242
Tree_Dataset\Field33	1269	71
Tree_Dataset\Field34	3700	200
Tree_Dataset\Field35	800	40
Tree_Dataset\Field36	1788	152
Total	69607	8713

them suitable for tasks such as object detection, where standardized coordinates are often used.

In object detection, the YOLO format is widely utilized for annotating and training models. This paper discusses mango orchard annotations in the YOLO format for tree detection tasks. This involves normalizing bounding box coordinates and structuring annotation lines in a standardized manner, ensuring compatibility with YOLO-based training methodologies. Bounding box coordinates are normalized to the values between 0 and 1. This is achieved by dividing the x -coordinate of the bounding box center (x_{center}) by the image width and the y -coordinate of the bounding box center (y_{center}) by the image height. Similarly, the width and height of the bounding box center (x_{center}) by the image width and the y -coordinate of the bounding box center (y_{center}) by the image height. Similarly, the width and height of the bounding box are normalized by dividing them by the image width and height. The YOLO annotation line is structured in the following format:

`< class >< xcenter >< ycenter >< width >< height >`

The class parameter is set to 0 since only one class is in the data. The structured YOLO annotation line containing information about the object's class, normalized coordinates of its center, normalized width and normalized height is a foundational component for training robust object detection models. This standardized annotation approach enhances the model's predictive accuracy and contributes to its adaptability to diverse visual datasets. The text file content generated for one annotated image is

0	0.098633	0.753906	0.197266	0.277344
0	0.683105	0.750977	0.321289	0.310547
0	0.731445	0.165527	0.326172	0.331055
0	0.136230	0.161621	0.272461	0.323242

All values are rounded up to 6 digits. The YOLO annotation format "0" denotes the class (in this case, trees). The annotation line is saved to a text file with a filename based on the original image. The same is applied to a larger image with multiple trees. This overall automatic annotation process generates two types of Excel files to organize and summarize the annotated data. The first type, stored within each

Table 5: Tree annotation details for the per image divided frames.

Annotated frames	Text file generated	Tree count
DJI_0172_frame_1_TP_1	true	3
DJI_0172_frame_2_TP_2	true	4
DJI_0172_frame_3_TP_3	true	4
DJI_0172_frame_4_TP_4	true	4
DJI_0172_frame_5_TP_5	true	4
DJI_0172_frame_6_TP_6	true	4
DJI_0172_frame_7_TP_7	true	4
DJI_0172_frame_8_TP_8	true	3
DJI_0172_frame_9_TP_9	true	3
DJI_0172_frame_10_TP_10	true	4
DJI_0172_frame_11_TP_11	true	4
DJI_0172_frame_12_TP_12	true	4
DJI_0172_frame_13_TP_13	true	3
DJI_0172_frame_14_TP_14	true	3
DJI_0172_frame_15_TP_15	true	4
DJI_0172_frame_16_TP_16	true	2
DJI_0172_frame_17_TP_17	true	3
DJI_0172_frame_18_TP_18	true	4
DJI_0172_frame_19_TP_19	true	4
DJI_0172_frame_20_TP_20	true	2

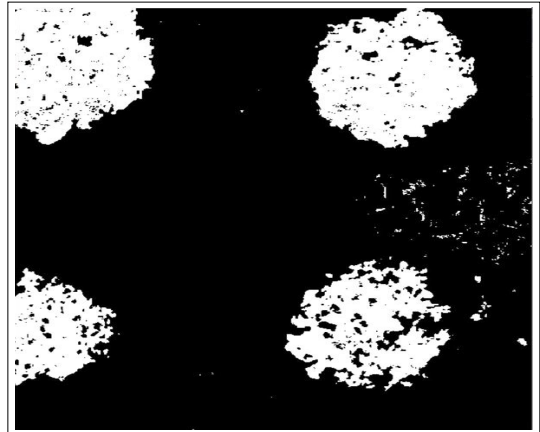


Fig 6: Binary image after applying HSV-based binary mask.



Fig 7: (a) Detected counters over the original image and (b) Bounding box drawn on the object with the area above threshold value.

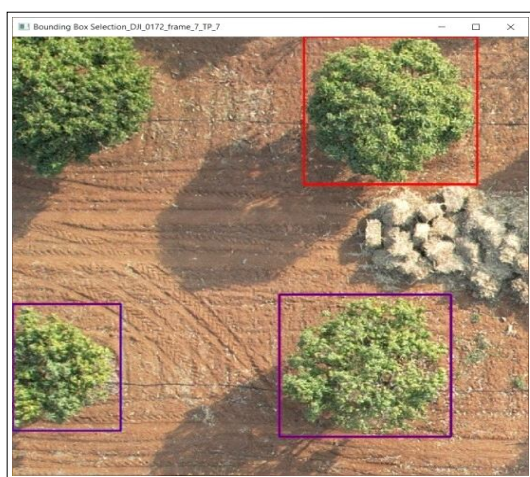


Fig 8: Bounding box confirmation with automatic tree detection.

Table 6: Tree present annotation count.

Annotated frames	Text file generated	Tree count
DJI_0172_frame_1_TP_1	true	3
DJI_0172_frame_2_TP_2	true	4
DJI_0172_frame_3_TP_3	true	4
DJI_0172_frame_4_TP_4	true	4
DJI_0172_frame_5_TP_5	true	4
DJI_0172_frame_6_TP_6	true	4
DJI_0172_frame_7_TP_7	true	4
DJI_0172_frame_8_TP_8	true	3
DJI_0172_frame_9_TP_9	true	3
DJI_0172_frame_10_TP_10	true	4
DJI_0172_frame_11_TP_11	true	4
DJI_0172_frame_12_TP_12	true	4
DJI_0172_frame_13_TP_13	true	3
DJI_0172_frame_14_TP_14	true	3
DJI_0172_frame_15_TP_15	true	4
DJI_0172_frame_16_TP_16	true	2
DJI_0172_frame_17_TP_17	true	3
DJI_0172_frame_18_TP_18	true	4
DJI_0172_frame_19_TP_19	true	4
DJI_0172_frame_20_TP_20	true	2

subdirectory under the base directory named as Tree_Dataset1\Field1\DJI_0172\DJI_0172_Annotation_Details.xlsx, includes columns for annotated frames (image filenames), a Boolean indicator for text file generation and the count of annotated trees per frame is shown in Table 5. This file structure provides detailed information about the annotated frames and their associated tree counts. The second type of Excel file, named Tree_Dataset1\All_Annotation_Details.xlsx, is stored in the root directory of the base directory and contains columns for the serial number, base directory name (representing the field), total annotated frames and the overall count of annotated bounding boxes across all subdirectories, which is shown in Table 6.

This main Excel file aggregates data from all subdirectories, offering a comprehensive overview of the entire dataset's annotation status, making it easier for researchers or analysts to access, analyze and visualize the annotated data effectively.

The Mango Orchard Aerial Image Dataset is publicly available at <https://www.kaggle.com/datasets/kavithahoude/mango-orchard-dataset> for researchers.

CONCLUSION

In this work the mango orchard images collected through drone are processed and YOLO format annotation dataset is created. The dataset includes the original images, frames obtained after segmentation and the annotated frames. The collected images are divided into smaller frames of sizes 1024×1024 in the preprocessing stage before applying machine learning and deep learning models. A process is developed to perfectly classify the frames two classes as frames with tree and without tree, which has achieved 100% accuracy. Then YOLO format annotations is computed with the user confirmation of the automatically added bounding box around the mango trees. We avoided selecting unwanted regions like tree shadows, other than mango trees, or grass in the annotation process. The frames are annotated to obtain the mango tree regions. The details of these trees are also provided in the Excel sheets for ease of use. This dataset will be made available for anyone willing to carryout research in the related problems. One of the potential applications can be pesticide spraying over the trees. As future enhancements in the annotation process, automatically added bounding box coordinates can be modified by including a confirmation option by the user. Further, other types of datasets such as Pascal VOC Format, COCO Format, TFRecord Format, LabelMe Format datasets can be prepared along with the segmentation dataset using advanced image processing techniques.

ACKNOWLEDGMENT

We would like to thank Dr. Visvesvarayya Hallur, Santosh Benni, Basavraj, University of Horticultural Sciences, Bagalkot(Karnataka) and Dr. Balasaheb Sawant Konkan Krishi Vidyapeeth, formerly Konkan Krishi Vidyapeeth, Dapoli, Ratnagiri, Maharashtra, for their assistance and support in data collection.

Disclaimers

The views and conclusions expressed in this article are solely those of the authors and do not necessarily represent the views of their affiliated institutions. The authors are responsible for the accuracy and completeness of the information provided, but do not accept any liability for any direct or indirect losses resulting from the use of this content.

Informed consent

All animal procedures are involved.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

REFERENCES

- Ausari, P.K., Gharate, P.S., Saikanth, D.R.K., Bijaya, O., Bahadur, R., Singh, Y.S. (2023). High-tech farming techniques in fruit crops to secure food demand: A review. *International Journal of Environment and Climate Change*. **13**: 2716-2730.
- Girijalaxmi, Houde, K.V. and Hegadi, R.S. (2024), In: Optimizing Drone Navigation using Shortest Path Algorithms Communications in Computer and Information Science, Springer Nature Switzerland, Cham. pp302-313.
- Goodchild, M.F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*. **69**: 211-221.
- Gulati, A. and Juneja, R. (2021). Innovations in production technologies in India. In: From Food Scarcity to Surplus, Springer. pp-23-82.
- Houde, K.V., Kamble, P.M. and Hegadi, R.S. (2024). In: Trees detection from aerial images using the YOLOv5 family Communications in computer and information science, Springer Nature Switzerland, Cham. pp314-323.
- Jose, C. and Jose, L. (2021). Crop monitoring using unmanned aerial vehicles: A Review. *Agricultural Reviews*. **42(2)**: 121-132. doi: 10.18805/ag.R-180.
- Jr, E., Doraiswamy, P.C., McMurtrey, J.E., Daughtry, C., Perry, E., Akhmedov, B. (2013). A visible band index for remote sensing leaf chlorophyll content at the canopy scale. *International Journal of Applied Earth Observation and Geoinformation*. **21**: 103-112.
- Khan, M.A., Ectors, W., Bellemans, T., Janssens, D., Wets, G. (2017). UAV-based traffic analysis: A universal guiding framework based on literature survey. *Transportation Research Procedia*. **22**: 541-550.
- Kumar, N., Singh, S.K., Reddy Obi, G.P., Mishra, V.N. and Bajpai, R.K. (2021). Remote sensing applications in mapping salt affected soils. *Agricultural Reviews*. **42(3)**: 257-266. doi: 10.18805/ag.R-2008.
- Ma, T., Zhou, C., Guo, H., Yang, G. and Zheng, H. (2019). Drone-based remote sensing for agriculture: A multisensor and multiresolution review. *ISPRS Journal of Photogrammetry and Remote Sensing*. **154**: 166-177.
- Mithra, S. and TYJ, N. M. (2021). A literature survey of unmanned aerial vehicle usage for civil applications. *Journal of Aerospace Technology and Management*. **13**: 1-22.
- Ray, S., Dadhwal, V. and Navalgund, R. (2022). Progress and challenges in earth observation data applications for agriculture at field scale in India and small farm holdings regions. *Journal of the Indian Society of Remote Sensing*. **50**: 189-196.
- Sawat, D.D. and Hegadi, R.S. (2016). Unconstrained face detection: A deep learning and machine learning combined approach. *CSI Transactions on ICT*. **5**: 1-5.
- Shakhatreh, H., Sawalmeh, A., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N.S., Khreishah, A., Guizani, M. (2019). Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access*. **7**: 48572-48634.
- Tuia, D., Marcos, D. and Camps-Valls, G. (2016). Multi-temporal and multi-source remote sensing image classification by nonlinear relative normalization. *ISPRS Journal of Photogrammetry and Remote Sensing*. **120**: 1-12.
- Vinita, V. and Dawn, S. (2022). Intuitionistic fuzzy representation of plant images captured using unmanned aerial vehicle for measuring mango crop health. In Fourteenth International Conference on Contemporary Computing, ACM, New York, NY, USA. 190-195.
- Zhang, L., Zhang, L. and Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*. **4**: 22-40.
- Zhu, X.X., Tuia, D., Mou, L., Gui-Song, X., Zhang, L., Xu, F., Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*. **5**: 8-36.