## I. Introduction

Agricultural automation has been at the forefront of technology innovations, with the goal of increasing orchard efficiency and productivity. One of the most critical tasks in this sector is recognising, counting, and localising fruits, all of which are required components in automated fruit-picking systems. Our project aims to use cameras as visual sensors to identify and count apples in an orchard, leveraging their ability to capture detailed images for processing with computer vision algorithms despite challenges.

Machine vision algorithms significantly enhance apple harvesting by accurately identifying and counting apples in orchards, overcoming environmental challenges and improving agricultural productivity and crop management.

While the combination of visual sensors and effective computer vision algorithms offers a potential option for automated apple counting in orchards, it is critical to handle the unique problems of the orchard environment. Our project aims to improve existing technologies and develop a highly reliable and effective agricultural automation system.

### *A. Aim and Objectives*

Our major goal is to create a robust system capable of detecting, counting, and localizing apples in orchard situations. To accomplish this, we have established particular goals aimed at addressing the inherent challenges of this application. These are some objectives:

1) Implementing a conventional image processing based approach and a machine learning approach algorithms that can perform efficiently in orchards in varying lighting circumstances.
2) To compare the accuracy and efficiency of two approaches and demonstrate the significant attention to detail that went into both algorithmic design and result analysis.

### *B. Challenges*

To implement apple counting algorithms utilizing image processing or deep learning approaches, it is essential to be aware of the challenges that may arise. These challenges are difficulty differentiating colours from an environmental view, such as differentiating between green apples and green leaves, variation in lighting conditions, and occlusion. Overcoming these will be highly instrumental in precise counting and localization. Consequently, a thorough review of recent plant and fruit detection methodologies from various studies is crucial to address these challenges.

## II. Related Works

### *A. Image Processing Approach*

Exploring edge detection methods shows tremendous potential in the field of apple detection and counting. The approaches found by Luo et al. (2021) [1] emphasise exploiting edge image processing and geometric morphology for improved detection accuracy. These approaches aim to precisely outline berry boundaries within images by using image pre-processing techniques such as noise reduction and contrast enhancement, followed by the application of advanced edge detection algorithms such as the Fast Radial Symmetry Transform (FRST) [2] and corner detection. This work is helpful since the proposed approach is meant to recognise round or quasi-circular grape berries, which may be translated to detect and count apples.

Based on the study conducted by Ganesan et al. (2017) [3], it was found that the Gaussian-based Canny Edge Detector is the most effective edge detection method for noisy conditions. This is due to its ability to incorporate Gaussian smoothing, which helps reduce noise such as uneven illumination and dense foliage and accurately distinguish apple and non-apple elements. Therefore, implementing a Canny Edge Detection algorithm can be a reliable solution for accurately identifying and detecting apples in an orchard environment.

### *B. Deep Learning Approach*

The methodology used in Yang et al. (2023) [4] involves the collection of image data of 2-year-old red Fuji apples from specific locations and the utilization of the YOLOv7 model for training and validation. The research tackles the problems created by high fruit density, occlusion, and overlapping in apple fruit target detection. Based on the YOLOv7 model, the paper provides an enhanced detection strategy for apple fruit targets, which tackles these problems and enhances the overall model's identification accuracy for complicated apple targets. The model's performance is evaluated using accuracy P, recall R, mean average precision (mAP), and F1 (metrics are defined in Section IV), which provides a full assessment of the proposed approach. The study does, however, admit challenges in reliably recognising apple targets with significant occlusion, emphasising the need for more enhancements to solve this difficulty.

## III. Data Acquisition and Datasets

Our research heavily relies on the MinneApple dataset, appropriately titled "MinneApple: A Benchmark Dataset for Apple Detection and Segmentation." [5] This dataset serves as a fundamental component, providing us with a rich and diverse collection of orchard images meticulously curated to support the development and evaluation of our orchard apple detection model. Originally gathered at the University of Minnesota Horticultural Research Center between 2015 and 2016, this dataset covers a critical period in apple growth and development, making it an invaluable resource.

The dataset includes images of apples at different stages of ripeness, ranging from unripe to fully ripe, allowing our model to handle the full spectrum of apple maturity. These apples exhibit various orientations and distances from the camera, replicating the diversity of their real-world distribution within orchards. Additionally, the dataset contains images with numerous other elements present in the scene, such as the sky, the ground, and abundant foliage. This complexity mirrors the cluttered and dynamic nature of orchard environments.

One notable feature of the MinneApple dataset is the presence of apples in both tree and ground locations. This distinction adds an extra layer of complexity to our counting methodology, as we aim to accurately count apples in different positions within the orchard. The ability to count apples on both trees and the ground is essential for practical orchard management and yield estimation, allowing us to comprehensively assess apple quantities across all orchard areas.

Furthermore, the dataset covers multiple orchards, each characterized by its unique environmental conditions. These orchards exhibit distinct lighting conditions influenced by factors such as weather, time of day, and seasonal fluctuations. The inclusion of diverse orchards ensures that our model can adapt to a wide range of real-world scenarios, making it a robust solution for apple counting in various orchard settings

The methodology employed in collecting this dataset was meticulous and involved the recording of video footage using a standard Samsung Galaxy S4 cell phone. The camera was strategically positioned horizontally, capturing a single side of a tree row during data collection. Subsequently, individual images were extracted from these video sequences, resulting in a comprehensive and diverse collection of orchard images.

To further enhance the dataset's diversity and bolster the robustness of our model, a series of meticulously chosen data transformations were systematically applied. These transformations were designed to simulate specific challenges commonly encountered in orchard environments, making the dataset a faithful representation of real-world conditions. Techniques such as rotation and flipping accounted for apples at different orientations and variations in orchard layouts. Scaling and zooming transformations introduced variations in camera-to-apple distances, providing diverse perspectives within the orchard. Adjustments to brightness and contrast faithfully mimicked changes in lighting conditions, addressing the dynamic nature of outdoor settings. Additionally, techniques like blurring, sharpening, shadow addition, and background variations were thoughtfully introduced to simulate various camera qualities, focus levels, lighting scenarios, and orchard landscapes, respectively.

By meticulously curating this dataset with a keen focus on diversity and real-world relevance and by strategically applying these data transformations, our goal was to equip our model with the adaptability and robustness needed for accurate apple detection in the face of the challenges posed by orchard environments. This dataset, combined with our model's training, serves as the cornerstone of our research, empowering us to develop a model that excels in real-world orchard scenarios.

These data transformations collectively contribute to the robustness and adaptability of our model. By exposing the model to a diverse set of scenarios during training, we aim to enhance its generalization capabilities, ensuring effective performance when deployed in actual orchard settings.

A careful selection of our diverse apple dataset and the strategic application of data transformations align with our goal of training a model capable of robust apple detection in challenging orchard environments. These considerations lay the foundation for a model that is not only accurate but also well-equipped to handle the complexities of real-world orchard scenarios.

## IV. Methodology

In our experiment, we present two approaches for counting apples in an orchard environment. The first approach utilizes conventional machine vision techniques, specifically employing the Canny algorithm and mean shift filtering, to detect apples based on their shape. We further enhanced this method to incorporate colour-based apple identification, utilizing the distinct characteristics of colour, shape, and other features [6].

The second approach involves a machine learning method, where we utilize the YOLOv7 approach. This technique leverages deep learning algorithms to accurately detect and count apples in the orchard.

### A. Approach A: Conventional Machine Vision Method

In the conventional machine vision approach, we start with image pre-processing; a blur method was applied to enhance the image quality and reduce noise. This involves using a Gaussian blur operation, which smooths out the image by convolving [Fig. 3] it with a Gaussian kernel. The blur helps reduce noise in the image and create a more uniform background. Additionally, mean shift filtering from the OpenCV library is applied to the blurred image. This technique groups pixels with similar colour or intensity values, effectively segmenting the image and enhancing regions of interest, such as the apple regions. The mean shift filtering algorithm from OpenCV identifies areas with similar colour characteristics and brings them closer together, resulting in better separation between objects. After mean shift filtering, the Canny edge detection algorithm, also from OpenCV, is employed to identify the boundaries of objects, specifically targeting the apple regions. This algorithm utilizes a multi-stage process to detect edges by calculating gradients and suppressing non-maximum values. The resulting edges highlight the distinctive features of the apple regions. A mask is utilized to further refine the detection and count only the apples. The mask is created by setting appropriate thresholds and selecting the regions of interest based on colour or intensity. By applying the mask, the algorithm focuses solely on the apple regions, disregarding other parts of the image. This enables an accurate count of the apples by considering only the relevant regions. The basic flow of the method is presented in [Fig. 4].

### B. Approach B: Machine Learning Method-YOLOv7

*YOLOv7 Configuration and Training:* In the YOLOv7 configuration phase, we delve into the specifics of the model's architecture, determining the number of layers and filters. YOLOv7 strikes a balance between computational efficiency and accurate predictions, utilizing convolutional layers, skip connections, and multi-scale features. This sophisticated structure enables the effective capture of nuanced details in orchard images. Embracing transfer learning, the model is initialized with pre-trained weights from the COCO dataset. This jump-starts its ability to recognize generic features, enhancing adaptability to diverse orchard conditions.

**Dataset Preparation:** Dataset preparation entails curating a diverse array of orchard images, intentionally incorporating variations in lighting, weather conditions, layouts, and distinct features of the background such as the sky and ground. This ensures a comprehensive representation of environmental factors that may impact image analysis algorithms. This diversity ensures the YOLOv7 model becomes robust in handling the complexities inherent in real-world orchard environments. The meticulous annotation process, involving bounding boxes and class labels, influences the model's capacity to accurately detect and count apples.

**Training the YOLOv7 Model:** Training includes fine-tuning hyperparameters, such as learning rates and batch sizes, to achieve optimal convergence without overfitting. The impact of transfer learning is carefully examined, expediting the learning process and enhancing the model's ability to recognize apples.

**Inference on Orchard Images:** YOLOv7's real-time processing capability is crucial for timely decision-making in precision agriculture. The model swiftly processes each orchard image, implementing probability thresholding to filter out less confident detections. This ensures only confident detections contribute to the final apple count.

**Post-processing for Counting:** Post-processing addresses overlapping bounding boxes through a merging algorithm, preventing double-counting in dense orchards. Spurious detections are removed, filtering out false positives that might have passed the initial probability threshold. This meticulous post-processing enhances the accuracy of the final apple count.

**Evaluation and Optimization:** Evaluating the YOLOv7 model's performance involves metrics such as precision, recall, and F1 score Section IV. These guide iterative optimization, which may include retraining with additional data, adjusting hyperparameters, or implementing strategies to address shortcomings. This iterative approach ensures continuous enhancement of the model's ability to detect and count apples across diverse orchard conditions.

The basic flow of the method is presented in [Fig. 1]. In the next section of the experiment and implementation, we will elaborate on each step for the two approaches.

### C. Evaluation Metrics

Evaluation metrics quantify the performance of a model, providing a standardized measure of its accuracy and effectiveness in solving a specific task. These metrics include accuracy, precision, recall, F1 score, depending on our problem.

**Count Accuracy:**

In the task of automating apple detection within an orchard environment, our primary metric for success was the accuracy of the apple count. This metric was quantified by comparing the algorithmically detected apple count to the manually verified ground truth count. The formula for calculating Count Accuracy is as follows:

$$\text{Count Accuracy} = \left(\frac{\text{Detected Count}}{\text{Ground Truth Count}}\right) \times 100$$

**Precision, Recall & F-1 Score:**
Precision, which is the proportion of actual apples in the total number of detections made by the model, indicating a high specificity in identifying true apples with a minimal rate of false identifications. The formula used for Precision is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

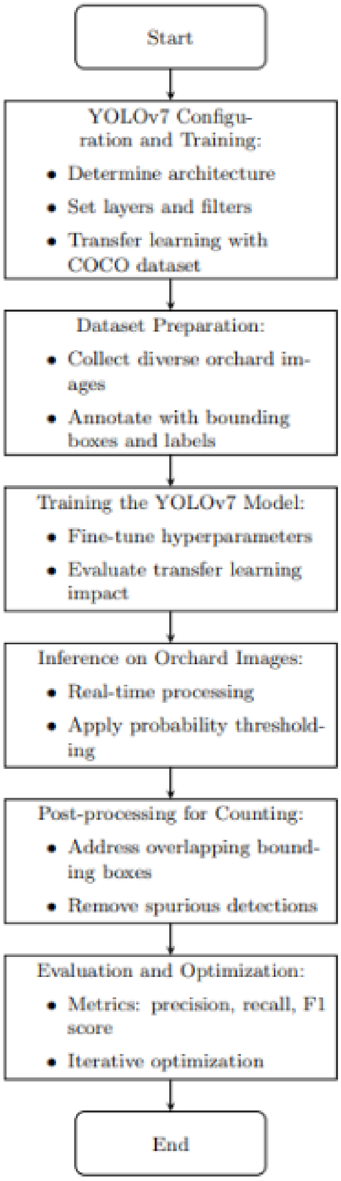

Start
YOLOv7 Configuration and Training:
• Determine architecture
• Set layers and filters
• Transfer learning with COCO dataset
Dataset Preparation:
• Collect diverse orchard images
• Annotate with bounding boxes and labels
Training the YOLOv7 Model:
• Fine-tune hyperparameters
• Evaluate transfer learning impact
Inference on Orchard Images:
• Real-time processing
• Apply probability thresholding
Post-processing for Counting:
• Address overlapping bounding boxes
• Remove spurious detections
Evaluation and Optimization:
• Metrics: precision, recall, F1 score
• Iterative optimization
End

Fig. 1. Detailed Methodology Flowchart of YOLOv7

The recall, representing the fraction of actual apples that were correctly identified from all the real apples available in the dataset. This metric underscores the model's sensitivity in recognizing apples that are present. The formula for Recall is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

The F1-Score, which harmonizes Precision and Recall, provides a singular measure of the model's overall accuracy by considering both the Precision and the Recall in its calculation. The formula for the F1-Score is:

$$\text{F1-Score} = 2 \times \frac{\text{Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}}$$

## V. Experiment and Implementation

### A. Approach A: Conventional Algorithm

The ground truth file path and the folder path containing the test images are specified. Variables for total apples and correct detections are initialized to 0. The ground truth data is loaded from the JSON file. The JSON file structure should be a dictionary with key-value pairs where the key represents the image name, and the value represents the number of apples in the picture. The JSON format for test data is referenced in Listing 1.

Listing 1: Sample groundtruth json used in program

```
# Structure of json used in the program.
#Note the data was taken from MineApple , data below is example data.
#Key Image name  and Value= Count of apples in the image(ground truth)
{
  "0.jpg": "1",
  "1.jpg": "1",
  "2.jpg": "4",
  "4.jpg": "21",
  "3.jpg": "3",
  "11.jpg": "4",
  "35.jpg": "18",
  "37.jpg": "10",
  "42.jpg": "2",
  "55.jpg": "17",
  "92.jpg": "0",
  .
  .
  .
  .
  .


  "94.jpg": "2",
  "100.jpg": "17",
1

}
```

Fig. 2. Sample groundtruth json used in program

A loop is then used to iterate over the files in the specified folder. For each image file, the image is read using cv2.imread(), and a copy of the image is created.

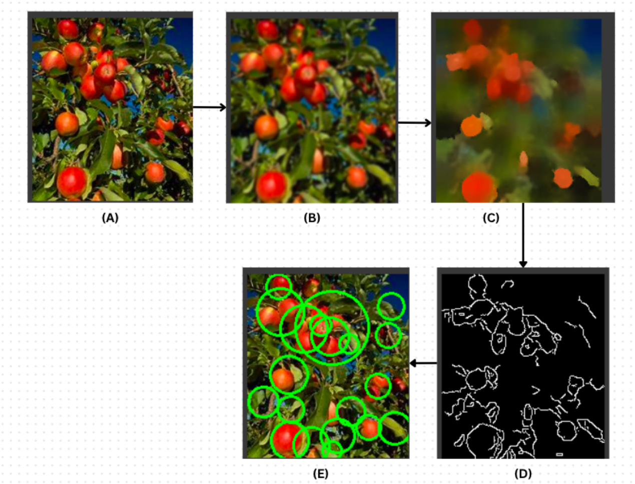


Fig. 3. Process Visualization: A) Original Image, B) Gaussian Blur, C) Mean Shift Filter, D) Canny Edge Detection, E) Contour and Counting

Image processing operations are applied to the image. Gaussian blur is performed using cv2.GaussianBlur() with a kernel size of (5, 5) to reduce noise. Mean shift filtering is then applied using cv2.pyrMeanShiftFiltering() [Fig. 4] to enhance the regions of interest. The Canny edge detection algorithm is applied using the auto_canny() function.

Contours are extracted from the edge image using cv2.findContours(). For each contour, a minimum enclosing circle is calculated using cv2.minEnclosingCircle(). If the radius of the circle is greater than 10, a circle and a text label with the contour number are drawn on the image.

The algorithm checks if the detected apple overlaps with the ground truth apple by comparing the filename with the ground truth data. If there is an overlap, the correct detection count is incremented.

The number of detected apples (c_num) is displayed, and the images (edge image and the image with circles and labels) are shown using cv2_imshow().

Finally, the accuracy is calculated by dividing the correct detection count by the total number of apples, and it is displayed as the dataset accuracy.

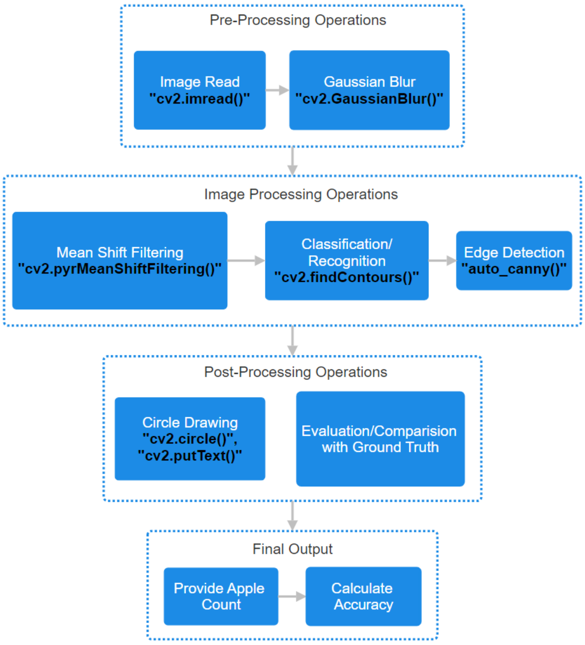

Pre-Processing Operations
Image Read "cv2.imread()"
Gaussian Blur "cv2.GaussianBlur()"
Image Processing Operations
Mean Shift Filtering "cv2.pyrMeanShiftFiltering()"
Classification/ Recognition "cv2.findContours()"
Edge Detection "auto_canny()"
Post-Processing Operations
Circle Drawing "cv2.circle()", "cv2.putText()"
Evaluation/Comparision with Ground Truth
Final Output
Provide Apple Count
Calculate Accuracy

Fig. 4. Conventional Machine Vision Approach flowchart.

## *B. Approach B: YOLOv7*

**Data Collection and Preprocessing:**

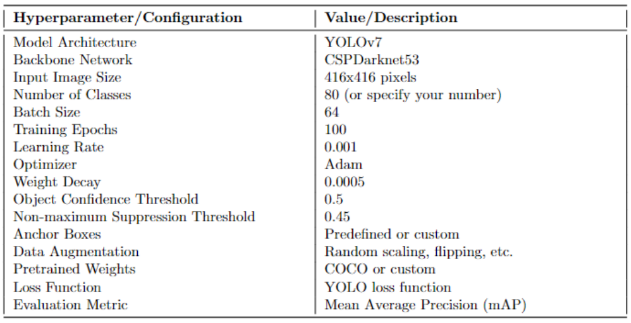
In the foundational phase of our research, the process of data collection and preprocessing played a critical role. We acquired the MinneApple dataset, which consisted of nearly 1000 low-resolution orchard images meticulously annotated with precise bounding boxes delineating individual apples. These annotations were carefully crafted by proficient annotators, serving as the essential ground truth for our object detection model.

To bolster the robustness of our model, we employed a range of data augmentation techniques. These included random rotations of up to ±15 degrees, controlled scaling between 0.85 to 1.15, brightness and contrast adjustments, and random flips. These augmentation strategies effectively simulated various real-world scenarios in orchards. Quality control measures were rigorously implemented to ensure dataset precision and diversity. Annotators adhered to guidelines and underwent evaluations to maintain fidelity, ultimately enabling accurate apple counting, even in challenging and diverse conditions.

**Model Configuration and Hyperparameter Tuning:**

The configuration of our YOLOv7 model and the meticulous tuning of hyperparameters were pivotal in achieving high accuracy in apple counting within orchard environments. As illustrated in Figure 5, we carefully selected several critical hyperparameters. The learning rate was set at 0.001, and we utilized a dynamic learning rate scheduler to adapt to evolving gradients and complex loss landscapes during the training process. Other key hyperparameters included a batch size of 16, 100 training epochs, momentum at 0.9, and weight decay at 0.0004. These choices were made to stabilize training and prevent overfitting, ensuring the model's robust performance.

Moreover, the selection of an input image size of 416x416 pixels struck a well-calibrated balance between detection precision and computational efficiency. Our data augmentation strategies introduced controlled variations, while the Adam optimizer, a warm-up period of 100 iterations, and default hyperparameters collectively optimized training stability.



| Hyperparameter/Configuration | Value/Description |
|---|---|
| Model Architecture | YOLOv7 |
| Backbone Network | CSPDarknet53 |
| Input Image Size | 416x416 pixels |
| Number of Classes | 80 (or specify your number) |
| Batch Size | 64 |
| Training Epochs | 100 |
| Learning Rate | 0.001 |
| Optimizer | Adam |
| Weight Decay | 0.0005 |
| Object Confidence Threshold | 0.5 |
| Non-maximum Suppression Threshold | 0.45 |
| Anchor Boxes | Predefined or custom |
| Data Augmentation | Random scaling, flipping, etc. |
| Pretrained Weights | COCO or custom |
| Loss Function | YOLO loss function |
| Evaluation Metric | Mean Average Precision (mAP) |

Fig. 5. YOLOv7 Hyperparameters and Configurations

**Post-Processing Customization:**

The final refinement of our apple counting methodology focused on precision and interpretability. To achieve this, we implemented a multi-step post-processing approach. Initially, we set a confidence threshold at 0.6, effectively filtering out low-confidence detections. This ensured that only highly confident predictions were considered in the final count.

We further applied Non-Maximum Suppression (NMS) with a threshold of 0.45. NMS played a crucial role in removing redundant bounding boxes, resulting in precise and uncluttered representations of detected apples. This step was instrumental in eliminating duplicate detections and refining the count.

Lastly, we systematically iterated through the detected objects, targeting confidently classified "apple" objects. This meticulous approach guaranteed that only genuine apples were counted, effectively eliminating any false positives from the count.

For interpretability and validation purposes, we employed visualization techniques that overlaid bounding boxes on the original images. This visualization provided a clear representation of the apple counts, allowing us to visually assess the accuracy of our model's predictions and make any necessary adjustments.

## VI. Results and Evaluation

This section presents the results of the proposed approach to the Apple in Orchard data sets. The model was evaluated based on its ability to accurately count the number of apples, even in terms of lighting, occlusion, and overall environment. In our quest to automate apple counting in orchards, we employed two distinct approaches: a deep learning model utilizing YOLOv7 and a conventional method based on edge detection. The objective was to compare the evaluation metrics: Count accuracy and precision, recall & F-1 Score.

### *A. Conventional Edge Detection*

In our apple counting project, the traditional edge detection approach was thoroughly evaluated using quantitative indicators to determine its effectiveness and performance. These measures provide insightful information on accuracy, precision, recall, and overall effectiveness. The conventional approach, based on edge detection techniques, achieved a respectable accuracy. This method focused on identifying the boundaries of apples using edge detection algorithms and implementing a counting mechanism for quantification. The model marks the apple by a circle and prints out the count of apples.

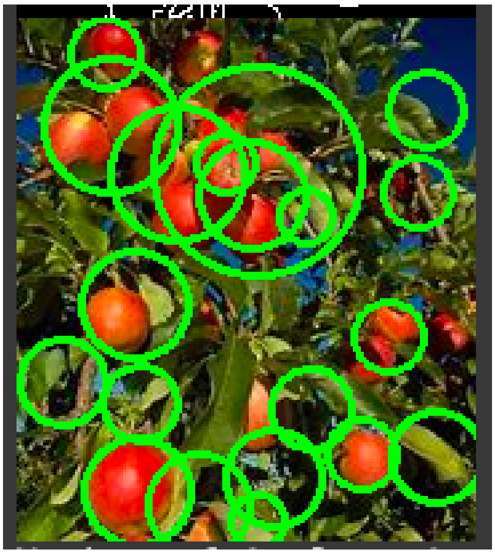


Fig. 6. Conventional method detection

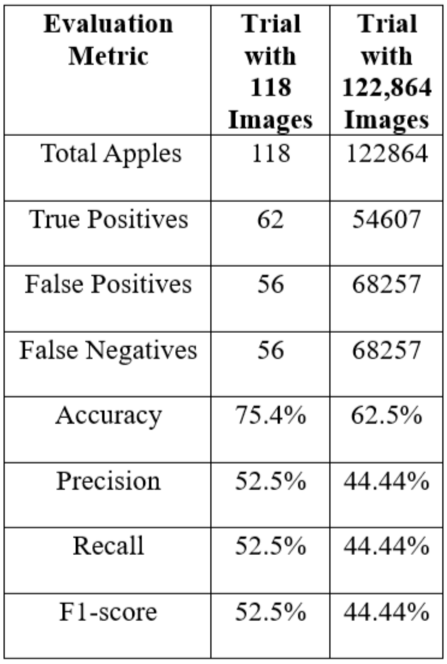
**Count Accuracy:** A vital criterion to assess the effectiveness of the traditional edge detection approach in executing apple recognition in orchard environments is count accuracy.

In the assessment stage, using the previously described count accuracy formula, the typical edge detection method's average count accuracy is calculated to be around 62.3% across the trials. This outcome highlights the method's accuracy in the detection phase and sheds information on how well it can count apples in orchard situations.

**Precision, Recall & F-1 Score:**

The accuracy rate of actual apples correctly detected by the conventional edge detection approach was 62.5%. This measure is essential for assessing how well the model can discriminate between real apple instances and other objects. With a precision value of 44.4%, the model's excellent specificity in identifying and properly classifying genuine apples is demonstrated by the low percentage of incorrect identifications. This accuracy number demonstrates the method's strength in diverse situations, as it is necessary for precise item recognition in real-world applications.

The percentage of true apples that were properly recognised out of all the truthful apples in the dataset was used



| Evaluation Metric | Trial with 118 Images | Trial with 122,864 Images |
|---|---|---|
| Total Apples | 118 | 122864 |
| True Positives | 62 | 54607 |
| False Positives | 56 | 68257 |
| False Negatives | 56 | 68257 |
| Accuracy | 75.4% | 62.5% |
| Precision | 52.5% | 44.44% |
| Recall | 52.5% | 44.44% |
| F1-score | 52.5% | 44.44% |

Fig. 7. Conventional Approach Output Metrics

to determine the Recall for the traditional technique, which was 44.4%. This indicator highlights the model's ability to detect apples that are currently in season.

Integrating Precision and Recall, the F1-Score was estimated to be around 44.4%. This unique gauge of total accuracy accounts for both recall and precision in its analysis.

The findings show that the traditional edge detection method maintains optimal performance, successfully integrating the ability to identify the majority of genuine apples in the test photos with the accuracy of detecting apples, with an F1-Score approaching 44.5%.

### B. Deep Learning with YOLO v7

The YOLO v7-based deep learning approach leveraged a comprehensive dataset, encompassing diverse orchard conditions and apple varieties. The training process involved fine-tuning the model to enhance its adaptability to the unique characteristics of our target orchards. The results of this model are explained by evaluation metrics.

The percentages shown in the [Fig. 8] are likely confidence scores. For each object it detects, it assigns a confidence score that represents the algorithm's certainty that the object has been correctly identified and localized. In the context of the [Fig. 8], which seems to depict apples on a tree, each percentage likely corresponds to how confident the YOLOv7 algorithm is that it has accurately detected an apple within the bounding box. Higher percentages mean the algorithm is more certain that the object is an apple. This kind of output is typical in machine learning models for object detection tasks.

**Count Accuracy:**

During our evaluation phase, with the average count, by applying the aforementioned formula, the Count Accuracy of our model is computed to be 82.46%.

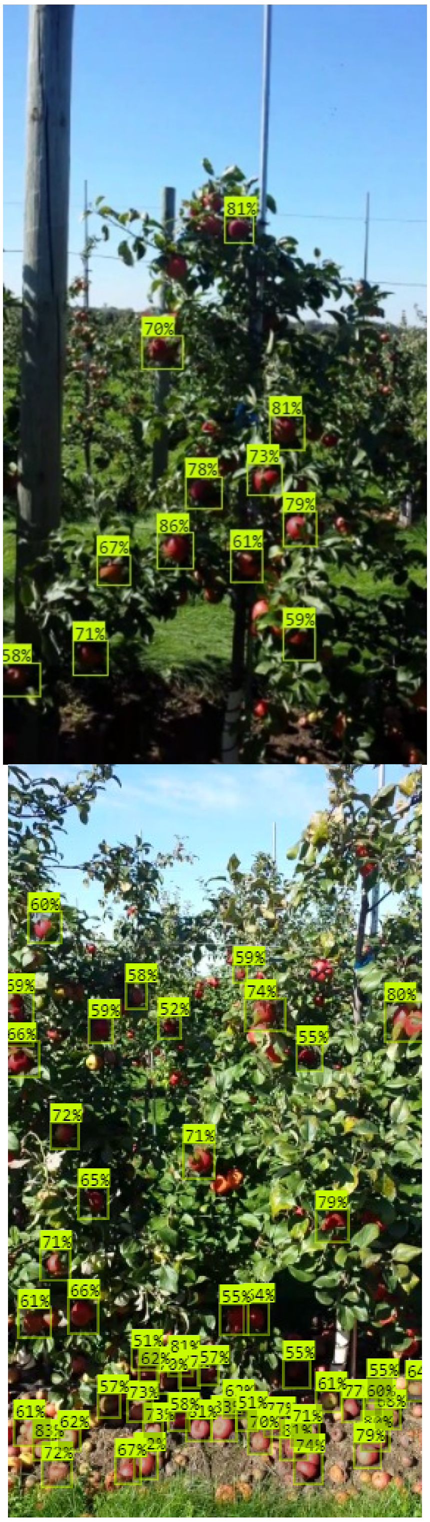
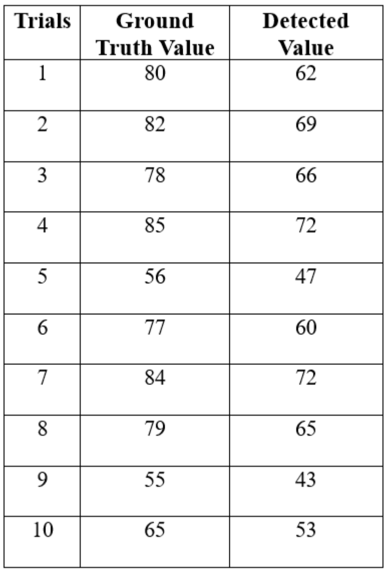


Fig. 8. Detection of Apples in Orchard using YOLO v7 model

This result signifies that our model's capability to accurately count apples stands at approximately 82.46%, indicating a high level of precision in the detection process. Such a level of accuracy showcases the model's efficacy, though it also highlights the room for improvement, particularly in the context of reducing false negatives and further aligning the detected count with the ground truth.

**Precision, Recall & F-1 Score:**

This model demonstrated a Precision of 92.6%, which is the proportion of actual apples in the total number of



| Trials | Ground Truth Value | Detected Value |
|---|---|---|
| 1 | 80 | 62 |
| 2 | 82 | 69 |
| 3 | 78 | 66 |
| 4 | 85 | 72 |
| 5 | 56 | 47 |
| 6 | 77 | 60 |
| 7 | 84 | 72 |
| 8 | 79 | 65 |
| 9 | 55 | 43 |
| 10 | 65 | 53 |

Fig. 9. Accuracy of Apple detection across different trials

detections made by the model, indicating a high specificity in identifying true apples with a minimal rate of false identifications.

The recall was calculated to be 84.7%, representing the fraction of actual apples that were correctly identified from all the real apples available in the dataset. This metric underscores the model's sensitivity in recognizing apples that are present.

The F1-Score, which harmonizes Precision and Recall, was computed to be approximately 88.47%. It provides a singular measure of the model's overall accuracy by considering both the Precision and the Recall in its calculation.

With an F1-Score nearing the 90% mark, the results indicate that the model maintains a balanced performance, effectively combining its ability to accurately detect apples with its capability to identify the majority of the actual apples in the test images. This balance is essential for practical applications where both the accurate and comprehensive detection of objects are critical for success.

**mAP:** This is a performance measure for object detection models. It represents the average precision across all classes and/or over different IoU (Intersection over Union) thresholds. Precision, in this context, refers to the model's ability to identify only the relevant objects in the image. The mean Average Precision is a common metric used to evaluate the accuracy of object detectors like YOLO, SSD, or Faster R-CNN.

Upon deployment of our model, it was trained and tested on a set of images with mAP observed:

Mean Average Precision (mAP): 90.8%

The graph in [Fig. 10] illustrates the evolution of the model's mean Average Precision (mAP) across training epochs. Initially, the model shows a lower mAP, typical of early training phases with weights that are yet to be optimized. As training proceeds, a notable increase in mAP is observed, evidencing the model's enhanced object detection capabilities. The mAP fluctuates due to the learning process's inherent variability but eventually starts to plateau, indicating that the model is reaching optimal performance. This plateau can signal when to cease training to avoid overfitting. Overall, this graph is a crucial diagnostic tool, guiding the adjustment of training duration and hyperparameters to refine the model's performance.

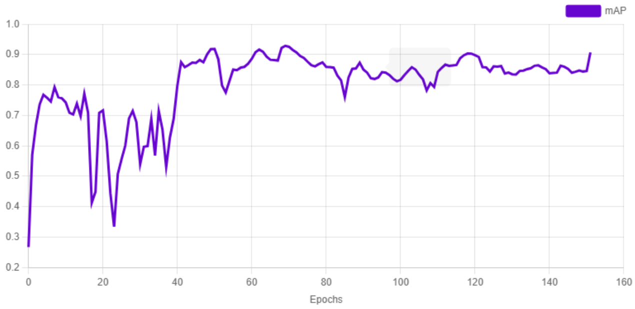
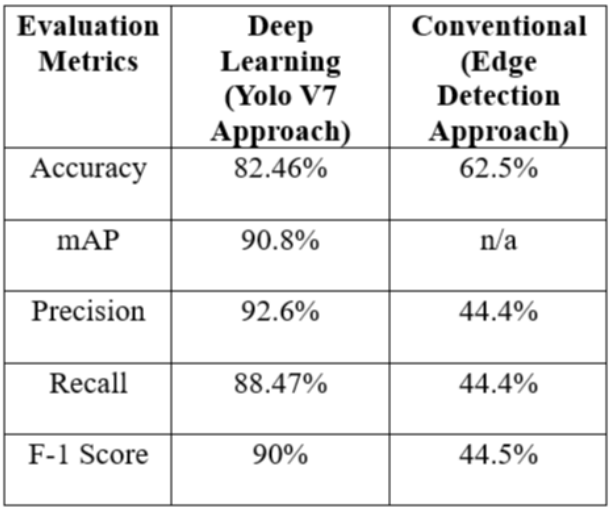


Fig. 10. Mean Average Precision

## VII. Conclusion and Future Works

In the table, The YOLO approach demonstrates superior performance compared to the conventional machine vision approach in detecting objects, specifically apples, in this scenario. Based on the table provided in [Fig. 11], the YOLO approach achieves an accuracy of 82.46%, precision of 92.6%, recall of 88.47%, and an F1 score of 90%. On the other hand, the conventional machine vision approach, utilizing the Canny edge detection algorithm in conjunction with an enhanced masking algorithm that incorporates colour masking, achieves an accuracy of 62.5%, precision of 44.4%, recall of 44.4%, and an F1 score of 44.5%.

In the conventional approach, the Canny edge detection algorithm with Gaussian blurring was initially employed to address edge detection. However, to overcome challenges related to colour differentiation and improve apple identification, an enhanced masking algorithm incorporating colour masking was implemented. This approach aimed to isolate and highlight the apple regions based on colour information.



| Evaluation Metrics | Deep Learning (Yolo V7 Approach) | Conventional (Edge Detection Approach) |
|---|---|---|
| Accuracy | 82.46% | 62.5% |
| mAP | 90.8% | n/a |
| Precision | 92.6% | 44.4% |
| Recall | 88.47% | 44.4% |
| F-1 Score | 90% | 44.5% |

Fig. 11. Detection of Apples in Orchard using YOLO v7 model

Despite the colour masking enhancement, the conventional machine vision approach falls short in terms of accuracy, precision, recall, and F1 score compared to the

YOLO approach. The YOLO approach's higher performance in accurately identifying and localizing apples while minimizing false positives and false negatives demonstrates its effectiveness in object detection tasks.

Future work can focus on further enhancing the conventional machine vision approach by exploring advanced feature extraction techniques, utilizing more sophisticated object detection algorithms, refining the colour masking approach, segmentation and collecting a larger and more diverse dataset. These improvements can help bridge the performance gap and enhance the accuracy, precision, recall, and F1 score of the conventional machine vision approach for apple detection.

However, the conventional approach demonstrated a different set of strengths, particularly in scenarios where apples were closely positioned. The evaluation metrics and analysis of false positives and false negatives provided a comprehensive understanding of the trade-offs inherent in each method.

In the next phase of our project, we are committed to further refining and enhancing the YOLO model's performance in counting apples in orchards. A major focus will be on data augmentation. We plan to significantly expand our dataset to include a wider range of images that represent diverse conditions. This expansion will include variations in lighting, apple sizes, and colours. By introducing these elements, we aim to improve the model's ability to generalize across different orchard environments.

Another key area of focus will be on fine-tuning the YOLO model's hyperparameters. We intend to experiment with various settings, such as the learning rate, batch size, and the number of epochs. This optimization process is critical to enhancing the overall performance of our model. In addition, we plan to explore advanced image preprocessing techniques. This will involve improving the quality of the input data through methods like noise reduction, contrast enhancement, and colour normalization, ensuring that the model receives the best possible input for accurate analysis.

We are also considering upgrading to the latest version of the YOLO model if it is not already in use. Newer versions often come with improvements that can boost detection accuracy. Complementing this, we will experiment with ensemble models. By combining the strengths of multiple model architectures, we hope to achieve better results than what might be possible with a single model approach.

## References

[1] L. Luo, W. Liu, Q. Lu, J. Wang, W. Wen, D. Yan, and Y. Tang, "Grape berry detection and size measurement based on edge image processing and geometric morphology," *Machines*, vol. 9, no. 10, p. 233, 2021.

[2] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, 2003.

[3] P. Ganesan and G. Sajiv, "A comprehensive study of edge detection for image processing applications," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–6, 2017.

[4] H. Yang, Y. Liu, S. Wang, H. Qu, N. Li, J. Wu, Y. Yan, H. Zhang, J. Wang, and J. Qiu, "Improved apple fruit target recognition method based on yolov7 model," *Agriculture*, vol. 13, no. 7, p. 1278, 2023.

[5] N. Haeni, P. Roy, and V. Isler, "Minneapple: A benchmark dataset for apple detection and segmentation," 2019. Retrieved from the Data Repository for the University of Minnesota, http://hdl.handle.net/11299/206575.

[6] S. Gajbhar, "Apple detection and counting," 2019. https://github.com/ShrishailSGajbhar/AppleDetectionAndCounting/tree/master.