small overlapped object detection and classification

Safaa Hdaiab
g202401120
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

***Abstract*—This research addresses the challenge of class imbalance and limited annotated data in multi-class object detection tasks for agricultural products, specifically date fruits. The proposed approach enhances detection performance by generating synthetic training data and improving the YOLOv11n model architecture.**

The methodology involves creating a synthetic image generation pipeline that samples individual fruit images (e.g., Ajwa, Medjool, Meneifi), applies random transformations, and places them on a shared canvas without overlap, while preserving YOLO-format annotations. This ensures balanced and diverse training samples to improve model generalization.

Additionally, the YOLOv11n architecture is modified by integrating attention mechanisms such as CBAM (Convolutional Block Attention Module) and novel blocks like C3k2 and C2PSA to enhance spatial and channel-wise focus. An ablation study is conducted to evaluate the impact of each architectural component on performance metrics like mAP, depth, and GFLOPs.

The expected outcome is a robust object detection model with improved accuracy, particularly for underrepresented date fruit classes, and better generalization in real-world agricultural environments

***Index Terms*—YOLOv11, Object Detection, Date Fruits, Synthetic Data Generation, CBAM, Attention Mechanisms, Class Imbalance, Agricultural AI, Data Augmentation, Deep Learning.**

I. Introduction

A. Background and Significance

The transformative potential of AI and ML in agriculture, offering innovative solutions to enhance fruit quality assessment, optimize harvesting decisions, and promote sustainable agricultural practices, thereby addressing critical challenges in food production and supply chain management [1]–[3]
Traditional methods of fruit classification and grading, especially for crops like date palm fruits and tomatoes, have relied heavily on manual inspection, which is labor-intensive, time-consuming, and prone to human error. This inefficiency poses challenges in meeting the increasing global demand for high-quality agricultural products and maintaining consistent quality standards in the supply chain [1], [4].

The development and application of AI-driven techniques, such as Convolutional Neural Networks (CNNs) and deep learning architectures, have emerged as powerful tools to automate and improve the accuracy of fruit classification and grading. CNNs enable the extraction of high-level features from images, facilitating the differentiation of fruit varieties based on physical attributes like color, size, shape, and texture without the need for manual feature engineering [1], [4]

In particular, the classification of date palm fruits is significant due to their nutritional, economic, and cultural importance in many regions. Accurate classification and grading of date fruits based on ripeness, quality, and variety are essential for optimizing harvest timing, improving market value, and reducing post-harvest losses. AI and ML methods provide a non-invasive, rapid, and scalable approach to address these needs, overcoming the limitations of traditional manual sorting [4], [5]

Moreover, the integration of AI with Internet of Things (IoT) technologies further enhances precision agriculture by enabling real-time monitoring and data-driven decision-making. This integration supports sustainable farming practices by optimizing resource use, reducing environmental impact, and improving crop yield and quality. The adoption of AI and IoT in agriculture is rapidly growing, with significant investments anticipated in intelligent, interconnected agricultural systems [3].

The significance of these advancements extends beyond date palms to other fruits such as apples, bananas, guavas, and tomatoes, where automated grading systems like FruitVision demonstrate high accuracy and efficiency, reducing labor costs and improving sorting precision on large-scale production lines [1], [6]

B. Challenges in Current Techniques

Current limitations in fruit classification, grading, and harvesting systems primarily stem from challenges related to manual processes, environmental variability, and the complexity of accurately detecting and classifying fruits in real-world conditions. These limitations motivate ongoing research and development to improve automation, accuracy, and robustness using advanced artificial intelligence (AI) and machine learning (ML) techniques.

Manual fruit classification and grading remain labor-intensive, time-consuming, and prone to human error, which affects consistency and efficiency in agricultural production. For example, date palm fruit sorting traditionally relies on visual inspection based on physical attributes such as color, size, shape, and surface texture, which is subjective and inefficient [4] Similarly, manual harvesting decisions depend on human judgment, which can be inconsistent and slow, limiting scalability [5]

Environmental factors such as varying illumination, natural backgrounds, and occlusions pose significant challenges for automated fruit detection and classification systems. Many existing methods, especially color-based and traditional machine learning approaches, struggle to maintain high accuracy under these variable conditions. For instance, many date maturity classification methods use images with uniform white backgrounds rather than natural orchard environments, limiting their practical applicability [5]. Likewise, conventional detection algorithms relying on shallow features like color and texture are sensitive to lighting changes and background complexity, reducing robustness [6]

Another limitation is the dependency of traditional machine learning methods on handcrafted features, which are often fruit-specific and require extensive domain knowledge and tuning. This restricts the generalizability of such systems across different fruit types and varieties [1]. Moreover, some deep learning models, while achieving high accuracy, can be computationally intensive and require large annotated datasets for training, which may not always be available [6].

The motivation for improvement arises from the need to develop more accurate, efficient, and scalable automated systems that can operate reliably in complex, real-world agricultural environments. Advances in deep learning, such as Convolutional Neural Networks (CNNs) and state-of-the-art object detection models like YOLOv5 and YOLOv8, offer promising solutions by enabling real-time, robust fruit detection and classification with minimal preprocessing [5], [6] These models can learn hierarchical features automatically, reducing the reliance on handcrafted features and improving adaptability across fruit types and conditions [4].

Furthermore, integrating AI with Internet of Things (IoT) technologies facilitates real-time data acquisition and decision-making, supporting precision agriculture and smart harvesting systems that optimize resource use and improve yield quality [3]. The development of comprehensive, annotated datasets captured under natural conditions also supports the training of more generalized and robust models [5].

C. Problem Statement

Accurately detecting and classifying different types of date fruits from real-world images is a challenging task due to variations in fruit appearance, background clutter, lighting conditions, and occlusion. Traditional object detection models often struggle to maintain high accuracy and generalization under such conditions, especially when dealing with a limited dataset.

This project addresses the problem of improving detection performance in such scenarios by enhancing the YOLOv11n architecture with an attention mechanism—Convolutional Block Attention Module (CBAM) [7]. The goal is to integrate spatial and channel attention into the detection pipeline to help the model better focus on relevant features, thereby improving its robustness and accuracy in identifying and localizing different varieties of date fruits in images.

D. Objectives

This project aims to develop an automated system for the detection and classification of date fruits using a deep learning pipeline designed to operate efficiently on local machines. By employing YOLOv11 (You Only Look Once, version 11) [8] for object detection, the system will accurately locate and crop date fruits from images—even in the presence of noisy backgrounds, occlusions, varying lighting conditions, or non-date objects. Once detected, the pipeline will classify the date fruits into different types (e.g., Ajwa, Medjool) using a robust classification model. To enhance feature representation and improve detection accuracy, the Convolutional Block Attention Module (CBAM) will be integrated into YOLOv11's convolution layers. This attention mechanism will enable the model to focus on the most informative parts of the image—such as the fruit region—while suppressing irrelevant background features. The system will be evaluated using standard metrics such as mean Average Precision (mAP), precision, recall, and F1-score to assess both detection and classification performance. Special emphasis will be placed on handling real-world dataset challenges, such as varying image resolutions, lighting conditions, and overlapping or occluded fruits. Designed to be lightweight yet accurate, the full pipeline

will be implemented and tested locally using Visual Studio Code and Anaconda, leveraging the power of an NVIDIA T1200 GPU for real-time inference. All development and experimentation will be documented and version-controlled using GitHub, ensuring reproducibility and facilitating collaboration. Ultimately, this work contributes to smart agriculture and food quality control by automating the classification and assessment of date fruits, supporting downstream tasks like packaging, quality grading, and pricing.

E. Scope of Study

The focus of this research is the development of a deep learning-based system for the automated detection and classification of date fruits in real-world images. The primary objective is to accurately identify and categorize various types of date fruits (e.g., Ajwa, Medjool) using state-of-the-art object detection and classification techniques, specifically leveraging the YOLOv11 architecture enhanced with attention mechanisms like the Convolutional Block Attention Module (CBAM). The research emphasizes improving the model's ability to handle complex backgrounds, occlusions, varying lighting conditions, and image quality—common challenges in agricultural image datasets.

The scope includes:

1) Designing a complete end-to-end pipeline combining object detection (YOLOv11) and classification for date fruit images.
2) Integrating CBAM into YOLOv11 to enhance spatial and channel-wise attention for better feature extraction and detection accuracy.
3) Evaluating model performance using standard metrics such as precision, recall, mAP, and F1-score.
4) Implementing the solution locally on a GPU-enabled system for real-time inference capability.
5) Addressing dataset variability and noise to ensure robustness in real-world agricultural settings.
6) Documenting and managing the project using GitHub for reproducibility and future extension.

This research contributes to smart agriculture by facilitating automated, accurate quality assessment of date fruits, supporting food supply chain applications such as grading, sorting, and packaging.

II. Literature Review

A. Overview of Existing Techniques

Existing techniques and models for fruit classification, grading, and harvesting have evolved significantly with the integration of artificial intelligence (AI), machine learning (ML), and deep learning (DL) methodologies, particularly convolutional neural networks (CNNs) and object detection frameworks.

B. Related Work

Traditional Machine Learning and Handcrafted Features: Several studies initially employed traditional ML techniques based on handcrafted features such as color, texture, shape, and size extracted from fruit images. For example, methods using Support Vector Machines (SVM), Artificial Neural Networks (ANN), and hybrid algorithms like ANN combined with Particle Swarm Optimization or Artificial Bee Colony have been applied for grading fruits such as bananas, oranges, guavas, and apples. These approaches rely on feature engineering, which is often fruit-specific and requires domain expertise, limiting their generalizability and robustness under varying environmental conditions [1].

Deep Learning and Convolutional Neural Networks (CNNs): Deep learning techniques, particularly CNNs, have become the dominant approach due to their ability to automatically extract hierarchical features from raw images without manual intervention. The FruitVision system, based on MobileNetV3 architecture pre-trained on ImageNet, exemplifies this trend by achieving high accuracy across multiple fruit types including apples, bananas, guavas, limes, oranges, pomegranates, dates, and mangoes. This model outperforms traditional ML methods and other state-of-the-art CNN architectures such as VGG19, ResNet, DenseNet, InceptionV3, and NASNetMobile [1]

Similarly, CNN models have been applied specifically for date fruit classification and grading, demonstrating high accuracy in distinguishing varieties and maturity stages. For instance, DateNET, a CNN architecture tailored for date palm fruit classification, uses layers of convolution, max-pooling, and dropout to process images and extract discriminative features [1].

Object Detection Models and Real-Time Applications: For fruit detection and harvesting, object detection models based on the YOLO (You Only Look Once) family have been widely adopted due to their real-time detection capabilities and high precision. YOLO treats object detection as a regression problem, enabling fast and accurate localization and classification of fruits in complex orchard environments. Variants such as YOLOv5 and YOLOv8 incorporate advanced backbone networks like Darknet-53 and utilize convolutional layers for feature extraction, neck modules for feature aggregation, and prediction heads for bounding box regression and classification. These models have been successfully applied to detect and classify fruits such as tomatoes and date palms under natural lighting and background conditions, facilitating automated harvesting decisions [5], [6]

Hybrid Approaches and Segmentation: Some studies combine deep learning with clustering or segmentation techniques to enhance maturity analysis and fruit detection. For example, the smart harvesting decision system for date fruits integrates YOLO-based detection with K-Means segmentation to classify fruit maturity stages by segmenting images into mature, immature, and background regions. This hybrid approach improves the precision of maturity classification in natural orchard settings [5].

Advanced CNN Architectures and Transfer Learning: The use of pre-trained CNN models such as EfficientNet, InceptionV3, ResNet, DenseNet, and MobileNetV3 leverages transfer learning to improve performance on fruit grading tasks with limited datasets. These architectures provide robust feature extraction capabilities and have been fine-tuned for specific fruit classification and grading problems, achieving accuracies often exceeding 90% and sometimes reaching above 99% . [1], [6]

C. Limitations in Existing Approaches

Dependence on Controlled Conditions and Limited Generalization;Many traditional and some current methods rely on images captured under controlled conditions, such as uniform white backgrounds or specific lighting setups, which limits their applicability in natural orchard environments. For example, color-based and machine learning approaches for date fruit maturity classification often use images taken post-harvest with uniform backgrounds rather than in-field images, reducing their robustness and practical utility for robotic harvesting systems [5]. Similarly, handcrafted feature-based machine learning models tend to be fruit-specific and struggle to generalize across different fruit types and environmental conditions [1].

Sensitivity to Environmental Variability;Traditional image processing and shallow feature-based methods are highly sensitive to variations in illumination, occlusions, and complex natural backgrounds. This sensitivity leads to decreased accuracy and reliability in real-world scenarios. For instance, conventional detection algorithms relying on color and texture features face challenges under varying lighting and background complexity, which is critical for outdoor fruit detection and harvesting [6].

Manual Feature Engineering and Computational Complexity;Machine learning approaches require extensive handcrafted feature extraction, which is time-consuming and demands domain expertise. This process limits scalability and adaptability. On the other hand, some deep learning models, while more accurate, can be computationally intensive and require large annotated datasets for effective training, which may not always be available or feasible in agricultural contexts [1], [6].

Limited Multi-View and Multi-Stage Analysis;Many current models, including the proposed FruitVision system, are trained and tested on single-view images of fruits, which may not capture the full variability of fruit appearance from different angles or stages of maturity. This limitation affects the robustness and accuracy of grading and classification systems in dynamic field conditions [1].

To address these limitations, there is a clear need for enhancement through the integration of attention mechanisms like the Convolutional Block Attention Module (CBAM). CBAM can significantly improve feature representation by allowing the network to focus selectively on important spatial and channel-wise cues. This enhancement not only improves detection accuracy but also helps maintain a lightweight architecture suitable for real-time applications [7].

In summary, the shortcomings of standard YOLOvv11 in handling real-world image variability and distinguishing fine-grained classes justify the proposed enhancement. Integrating CBAM is a strategic improvement that aligns with the goals of developing robust, accurate, and efficient detection systems for smart agriculture and food quality control.

III. Proposed Methodology

A. Existing Model and Challenges

Despite the advancements in deep learning for object detection and classification, existing models such as the baseline YOLOv8n still face significant challenges when applied to real-world agricultural datasets like those containing date fruits. One major shortcoming is the model's limited ability to focus on the most relevant parts of the image—particularly when the scene includes cluttered backgrounds, varying lighting conditions, occlusions, or overlapping objects. These limitations often result in reduced precision, missed detections, or incorrect classifications, especially for visually similar date fruit types such as Ajwa and Medjool.

Additionally, standard convolutional layers in YOLOv11n process spatial and channel information uniformly, without explicitly learning to prioritize more informative features over irrelevant ones. This becomes problematic when the background distracts the model or when critical distinguishing features are subtle and require enhanced attention.

B. Proposed Enhancements

Integration of CBAM into YOLOv11n: The primary innovation lies in augmenting the YOLOv11n architecture with the Convolutional Block Attention Module (CBAM). This attention mechanism is strategically embedded into the convolutional layers of the YOLOv8n backbone to refine feature representation by emphasizing salient spatial and channel-wise information. This modification allows the model to better focus on the date fruits and suppress background noise, significantly improving detection in complex real-world scenarios.

Customized YOLOv11n for Agricultural Use: The base YOLOv11n model is adapted specifically for date fruit detection and classification, where typical object detection challenges include variable lighting, occlusions, overlapping objects, and noisy backgrounds. This task-specific customization of the network architecture enhances both robustness and accuracy for agricultural datasets.

End-to-End Detection and Classification Pipeline: The research presents a full pipeline—from detection using the enhanced YOLOv11n+CBAM model to classification of date fruit types (e.g., Ajwa, Medjool) using a lightweight CNN classifier or fine-tuned pretrained model. This ensures that the system is capable of handling the complete workflow from raw input to labeled output.

Performance Evaluation and Benchmarking: The impact of attention-enhanced YOLOv11n is rigorously evaluated against the baseline version using standard object detection metrics such as mAP, precision, recall, and F1-score. These benchmarks validate the effectiveness of CBAM in improving detection and classification accuracy.

Locally Deployable Real-Time System: By maintaining YOLOv11n's real-time inference speed, the proposed system remains lightweight and computationally efficient. It is optimized for deployment on local machines, including those with limited GPU resources like the NVIDIA T1200, enabling practical use in agricultural environments or mobile applications.

Contribution to Smart Agriculture: This project contributes a scalable and accurate solution to automated fruit quality assessment, supporting smart agriculture initiatives by reducing manual effort in sorting, grading, and quality control within the supply chain.

Together, these contributions present a novel and practical solution that blends state-of-the-art object detection with attention-based refinement to advance agricultural image analysis.

C. Algorithm and Implementation

1. Algorithmic Steps of CBAM-YOLO11n Detection Pipeline

1.1 Model Initialization: The YOLO11n model is initialized with a custom architecture consisting of:

- **Backbone**: A series of convolutional layers and C3 modules to extract multi-scale features.
- **Neck**: Utilizes `SPPF`, `CBAM`, and `C2PSA` for enhanced feature fusion and attention.
- **Head**: Generates predictions at three different scales corresponding to small, medium, and large objects.

The CBAM module is integrated into the backbone after the final feature extraction stage:

```
[-1, 1, CBAM, [1024]]
```

This enhances the model's ability to focus on informative regions via channel and spatial attention mechanisms.

2. Dataset Description

The dataset used consists of labeled images of date fruits, such as:

- **Ajwa**
- **Medjool**
- **Meneifi**
- **Sukarie**
- **Nabat Ali**
- **Sheishi**
- **Sugey**

Each image has an accompanying YOLO-format annotation file:

```
<class_id> <x_center> <y_center> <width> <height>
```

3. Data Preprocessing Techniques

3.1 Image Synthesis Techniques

3.1.1 Motivation for Synthetic Images: Deep learning models, particularly in object detection tasks, require large amounts of labeled data to achieve robust performance. In scenarios where collecting real-world annotated images is difficult, time-consuming, or costly, synthetic image generation offers a powerful alternative. It enables the expansion of the dataset by artificially creating new training samples that closely resemble real-world data.

3.1.2 Synthetic Image Generation Approach: In this project, synthetic images were generated to enhance the dataset diversity and to improve the generalization ability of the CBAM-YOLO11n detection model. The following techniques were utilized:

- **Object Overlaying**: Individual date fruit instances were cropped from existing images and then programmatically pasted onto varied random or clean

background images. This simulates different real-world environments.

- **Random Scaling and Rotation**: Synthetic objects were resized and rotated to introduce variability in shape and orientation, which helps the model learn rotation and scale-invariant features.
- **Background Blending**: Blending techniques (e.g., alpha blending or Gaussian smoothing) were applied at the boundaries to make overlaid objects appear more naturally integrated into the scene.
- **Synthetic Lighting and Shadows**: To replicate natural light conditions, simple shadow masks and lighting filters were applied to synthetic images. This enhances robustness against illumination changes.

3.2 Automatic Annotation Generation: One key advantage of synthetic image generation is the ability to automatically generate bounding box annotations. Since the exact placement, size, and class of each pasted object is known, the corresponding YOLO-format annotation can be created programmatically, reducing annotation overhead and ensuring 100% accurate labels.

3.3 Benefits in Model Performance: Incorporating synthetic images improved the model's ability to detect date fruits in diverse environments. This is attributed to:

- Improved data diversity, leading to better generalization.
- Increased sample count, which helps reduce overfitting.
- More balanced representation of underrepresented classes.

3.4 Tools and Libraries Used: Synthetic images were generated using a combination of:

- **Python Imaging Library (PIL)** and **OpenCV** for image manipulation.
- **Albumentations** for applying advanced image augmentations.
- Custom Python scripts to automate pasting, blending, and annotation generation.

3.5 Annotation Conversion: Annotations are formatted to YOLO's expected structure with relative coordinates.

4. CBAM Attention Mechanism

4.1 Channel Attention:

- Applies global average pooling across spatial dimensions.
- Passes the result through a shared MLP to compute channel importance.

4.2 Spatial Attention:

- Computes mean and max across channels.
- Concatenates and applies a 7x7 convolution to get spatial attention.

4.3 Combined Attention: CBAM applies attention sequentially, first through channel attention and then spatial attention. The overall output is computed asI:

$$\begin{aligned} \text{CA}(x) &= \text{ChannelAttention}(x) \cdot x \\ \text{Output} &= \text{SpatialAttention}(\text{CA}(x)) \cdot \text{CA}(x) \end{aligned} \quad (1)$$

5. Training Pipeline

- **Loss Function**: Combines objectness, classification, and bounding box regression losses.
- **Optimizer**: SGD or Adam with cosine learning rate schedule and warm-up.
- **Metrics**: Mean Average Precision (mAP@0.5), Precision, Recall.

6. Postprocessing

- Non-Maximum Suppression (NMS) is used to filter overlapping bounding boxes.
- Output boxes are rescaled to original image dimensions.

D. Loss Function and Optimization

We used a customized loss setup for training the YOLOv11n model with CBAM modules and the SIOU loss function. The components are described II:

1. Attention Module Effects

Two CBAM layers were introduced. However, the mAP for the `Ajwa` class dropped despite it having the most instances.

1.1 Observation: CBAM may suppress features relevant to `Ajwa`, due to:

- Global average pooling losing class-specific signals
- Uniform channel/spatial attention suppressing dense features

1.2 Response Actions:

- Increased box loss weight: `box=0.1`
- Increased classification loss weight: `cls=0.4`, $\gamma = 2.0$
- Proposed label smoothing to mitigate confident misclassifications
- Enabled warmup epochs: `warmup_epochs=2.0`
- Suggested attention map visualization for debugging

2. Optimization and Regularization

3. Ajwa Class mAP Drop: Hypothesis

Even with high sample count, CBAM may suppress `Ajwa`-specific signals:

- Overlapping features with other classes
- Insufficient separation by attention mechanism

Step	Description
Model	Custom YOLOv11n with CBAM in the backbone.
Dataset	3-class date fruit detection: Ajwa, Medjool, Meneifi.
Image Preprocessing	Resize to 640×640, normalize, and augment.
Annotations	YOLO format: `class_id x_center y_center width height`.
Attention	CBAM: channel and spatial attention blocks.
Training	YOLO loss with cosine LR schedule and warm-up.
Output	Bounding boxes with class probabilities after NMS.

TABLE I: Pipeline summary for YOLOv11n with CBAM integration.

Loss Component	Configuration	Purpose
Box Loss	SIOU, `box=0.05`	Captures angle and shape mismatch for localization
Class Loss	Focal Loss (`cls=0.3`, $\gamma = 1.5$)	Reduces imbalance and emphasizes hard examples
Distribution Loss	`dfl=1.0`	Anchor-free bounding box regression enhancement

TABLE II: Summary of Loss Components in YOLOv11n Training

Feature	Setting	Purpose
Weight Decay	0.0002	Regularization to prevent overfitting.
Auto Augment	RandAugment	Improves generalization across samples.
Mixup	Off	Consider enabling for handling class confusion.
Dropout	0.0	Suggest 0.1 if overfitting is observed.
Determinism	Enabled	Ensures reproducibility of training results.

TABLE III: Training regularization and optimization features.

4. Next Steps

1) Implement class-aware CBAM or class-conditioned attention
2) Enable `mixup=0.2` to increase intra-class variability
3) Log loss and attention maps per class
4) Run ablation study on CBAM positioning and depth

IV. Experimental Design and Evaluation

A. Datasets and Preprocessing

The dataset utilized in this study comprises annotated images depicting various types of date fruits, including but not limited to Medjool and Meneifi,Ajwa,Sukarie, Nabat Ali..etc . Each image is paired with a corresponding annotation file in the YOLO (You Only Look Once) format, which encapsulates the object class and bounding box coordinates in a normalized structure. The primary objective of this dataset is to enable the training of object detection models that can accurately localize and classify date fruit varieties within natural images. The dataset is organized into separate directories for training, validation, and testing, each containing image files (.jpg) and their corresponding annotation files (.txt). The annotation files adhere to the YOLO format, where each line represents an object as a tuple consisting of the class index followed by the normalized bounding box coordinates

`(x_center, y_center, width, height)`

Additionally, a classes.txt file enumerates all the class labels in the dataset, ensuring consistency across annotations.

Preprocessing Pipeline: To facilitate efficient model training and enhance generalization capabilities, the following preprocessing techniques were applied:

B. Performance Metrics

In this project, several performance metrics were used to evaluate the effectiveness of the YOLOv11vn model with CBAM attention modules and the SIoU loss function for the task of date fruit classification. Below are the key metrics and their significance in this context:

C. Intersection over Union (IoU)

IoU is a fundamental metric in object detection used to measure the overlap between the predicted bounding box and the ground truth box. Mathematically, it is defined as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

A higher IoU indicates a better fit between the predicted and actual object locations. In this project, a threshold of 0.7 was used, making the detection criteria more strict to promote high spatial accuracy.

D. Mean Average Precision (mAP)

The mAP metric is commonly used in object detection tasks to summarize the precision-recall curve. In this project, the mAP@IoU[.50:.95] was utilized, which averages the precision across multiple IoU thresholds from 0.5 to 0.95 (step 0.05), providing a robust measure of overall performance.

E. Class-wise mAP

Given the imbalanced dataset (e.g., the *Ajwa* class had significantly more instances), class-wise mAP was critical in analyzing how each fruit variety was detected. Interestingly, the addition of CBAM attention layers resulted in a decrease in mAP for the *Ajwa* class, suggesting potential suppression of dominant class features due to the attention mechanism.

F. Attention Map Visualization (Qualitative Metric)

Although not a quantitative metric, attention maps from the CBAM layers were logged to understand how the model distributes its focus across spatial and channel dimensions. The mean values of the attention outputs hovered around 0.5, which may indicate uniform or overly saturated attention maps, potentially reducing their effectivenessIV.

G. Experiment Setup

V. Experimental Setup and Configurations

This section outlines the experimental framework, including model architecture, training configurations, dataset characteristics, and augmentation strategies used in our project.

A. Model Architecture

The backbone of the object detection model was a custom-modified YOLOv11vn architecture enhanced with two CBAM (Convolutional Block Attention Module) layers. These were strategically placed after the P3 feature extraction and after the SPPF module to improve feature representation through both spatial and channel attention.

- **Backbone:** YOLOv11vn with modified residual blocks (C3k2) and SPPF.
- **Attention Modules:** Two CBAM layers introduced at mid and high-level feature maps.
- **Loss Function:** SIoU (Skew Intersection over Union) used for bounding box regression, which better handles object aspect ratios and alignment compared to standard IoU.

B. Dataset

The dataset used consists of labeled images of various date fruit classes (e.g., Ajwa, Medjool, Meneifi), annotated in YOLO format. The dataset was split into training and validation sets as defined in `dataset.yaml`.

- **Image Size:** 640x640 pixels
- **Classes:** Multiple date fruit varieties
- **Data Split:** Training and validation via `split: val`

C. Training Configuration

The model was trained using the Ultralytics YOLO framework with the following settings:

- **Epochs:** 50
- **Batch Size:** 16
- **Optimizer:** Auto-selected (typically Adam or SGD with momentum)
- **Learning Rate:** Initial = 0.005, final = 0.1 (cosine decay disabled)
- **Loss Weights:** `box = 0.05`, `cls = 0.3`, `dfl = 1.0`, `pose = 12.0`
- **Device:** CUDA-enabled GPU (device 0)
- **Determinism:** Set to True to ensure reproducibility
- **Caching:** Enabled to speed up training

D. Data Augmentation and Regularization

To improve model generalization and robustness, the following augmentations were applied:

- **Flip:** Horizontal flips with 0.5 probability
- **HSV Shifts:** Applied to hue (0.0138), saturation (0.664), and value (0.464)
- **Translation and Scaling:** Minor translation (0.05) and scaling (0.2)
- **Erasing:** Random erasing with 0.4 probability
- **Auto Augmentation:** Enabled using `randaugment`
- **Mosaic:** Enabled with a probability of 0.5 for synthetic image creation

E. Validation Strategy

- Validation was conducted after every epoch using the `val` split.
- Performance was evaluated using mAP@0.5:0.95, IoU thresholds, and class-wise breakdown.

This well-rounded configuration was aimed at maximizing detection performance, especially under varying lighting and occlusion conditions typical in real-world fruit datasets.

F. Results Comparative Analysis

Yolo with attention model CBAM: After integrating CBAM (Convolutional Block Attention Module) into the YOLOv11n model, the performance metrics show notable improvements in certain areas, especially for

Metric	Purpose	Observation
IoU (threshold = 0.7)	Localization accuracy	High threshold encourages precise bounding boxes.
mAP@[.5:.95]	Overall detection performance	Used to compare models with and without CBAM.
Class-wise mAP	Per-class detection quality	*Ajwa* mAP dropped with CBAM despite data abundance.
Attention Map Stats	Qualitative interpretability	Channel/Spatial mean $\sim$0.5 suggests saturation.

TABLE IV: Summary of performance metrics and their relevance to the project.

minority classes. Overall, the model achieved a box precision (P) of 0.928 and a recall (R) of 0.566, with an mAP50 of 0.662 and an mAP50-95 of 0.507. This indicates that while the model is highly precise when it predicts a box, it still misses a number of objects during detection (moderate recall).

Looking into the individual classes, Ajwa, despite being the most abundant class (143 instances), shows high precision (0.934) but a very low recall (0.0909), resulting in a mAP50 of 0.231 and a mAP50-95 of 0.108. This suggests that even with CBAM, Ajwa detections are highly confident but rarely triggered. Medjool, similarly, achieves strong precision (0.97) but weak recall (0.152), leading to a modest mAP.

On the other hand, CBAM significantly benefits the minority classes. Meneifi achieves a precision of 0.919 and a much higher recall of 0.669, with a strong mAP50 of 0.876 and mAP50-95 of 0.622. Nabtat Ali and Shaishe perform even better, with Nabtat Ali reaching perfect precision (1.0), high recall (0.88), and an excellent mAP50 of 0.984. Notably, Sugaey, the rarest class, attains a recall of 1.0 and a very strong mAP50-95 of 0.805, indicating the model's ability to confidently detect rare classes without missing any instances. background

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	100	281	0.928	0.566	0.662	0.507
Ajwa	24	143	0.934	0.0909	0.231	0.108
Medjool	10	46	0.97	0.152	0.232	0.193
Meneifi	17	17	0.919	0.669	0.876	0.622
Nabtat Ali	9	9	1	0.88	0.984	0.761
Shaishe	15	15	0.854	0.8	0.85	0.671
Sokari	18	44	1	0.369	0.484	0.393
Sugaey	7	7	0.816	1	0.978	0.805

Fig. 1: CBAM (Convolutional Block Attention Module) into the YOLOv11n model

misclassification dominates the matrix, especially for Ajwa and Medjool. This supports the earlier observation that although CBAM helped enhance Comparing the baseline model to the CBAM-enhanced model shows clear improvements, especially in challenging areas. Initially, the baseline achieved an overall precision of 68.3%, recall of 70.5%, mAP50 of 72.5%, and mAP50-95 of 48.0%. After introducing CBAM, precision rose sharply to 92.8%, while recall slightly decreased to 56.6%. Most notably, mAP50 improved to 66.2% and mAP50-95 to 50.7%, suggesting that CBAM helped the model localize and classify objects more precisely, even if it missed some harder examples.

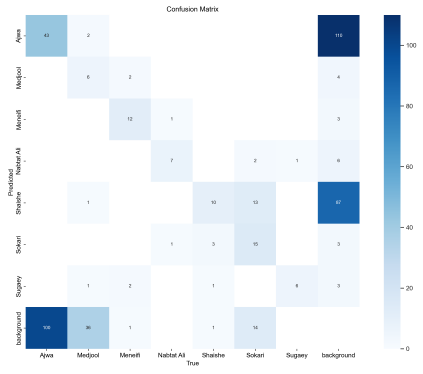

Fig. 2: CBAM (Convolutional Block Attention Module) into the YOLOv11n model

On a per-class level, classes like Meneifi and Shaishe greatly benefited: Meneifi's mAP50 increased from 81.3% to 87.6%, and Shaishe rose from 47.1% to 85.0%. Sugaey also improved its already good performance, pushing mAP50-95 from 45.5% to 80.5%. However, some classes such as Ajwa still struggled, achieving only 23.1% mAP50 after CBAM, compared to 19.7% before — meaning that CBAM alone isn't enough to fully address class imbalance or heavy background confusion. In summary, CBAM significantly improved precision,

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 10(
all	70	243	0.683	0.705	0.725	0.48
Ajwa	10	86	0.415	0.14	0.197	0.086
Medjool	41	45	0.754	1	0.957	0.591
Meneifi	13	15	0.815	0.6	0.813	0.52
Nabtat Ali	22	22	0.767	0.955	0.955	0.67
Shaishe	13	30	0.526	0.467	0.471	0.286
Sokari	23	30	0.908	0.984	0.985	0.755
Sugaey	10	15	0.597	0.789	0.694	0.455

Fig. 3: yOLOv11n model

per-class localization (mAP50-95), and class separation across most categories, but recall and background confusion still need further attention. A next good step could be combining CBAM with CBFocalLoss or data balancing techniques to push both recall and mAP higher.

VI. Extended Contributions

VII. Broader Impact

This research presents significant contributions not only to the field of computer vision and deep learning but also to real-world applications in agriculture, food quality control, and automation.

Feature	mAP Impact	Depth Impact	GFLOPs Impact	Notes / Usefulness
C3k2 Block	Moderate ↑	High ↑	High ↑	Increases representational power; useful for complex feature learning; risk of overfitting on small datasets.
CBAM (Attention)	1–3% ↑	Medium ↑	Medium ↑	Improves focus on important regions; enhances generalization in cluttered or noisy scenes.
C2PSA Block	Moderate ↑	Medium ↑	Medium ↑	Adds spatial awareness and long-range interaction; helpful for overlapping or small objects.
SPPF	Small ↑	Balanced	Efficient ↔	Enlarges receptive field without much cost; supports multi-scale detection.
Upsample + Concat (FPN)	Strong ↑	Minimal ↓	Medium ↑	Critical for multi-scale detection and context fusion; enhances performance across object sizes.
Increased Network Depth (364 Layers)	Dataset-dependent	High ↑	High ↑	Boosts model capacity; may overfit or cause vanishing gradients if not properly trained.

TABLE V: Impact Analysis of Proposed YOLOv11n Features.

A. Agricultural Automation

The deployment of lightweight, accurate object detection models like YOLOv11vn with CBAM can revolutionize the agricultural industry by enabling automated classification and quality assessment of fruits. For example, precise detection and differentiation between date fruit varieties such as Ajwa, Medjool, and Meneifi can aid in:

- **Automated Sorting:** Enhancing the speed and accuracy of fruit sorting lines in packaging facilities.
- **Yield Estimation:** Facilitating yield monitoring and grading in farms using drone or robot-mounted systems.
- **Labor Reduction:** Reducing reliance on manual inspection, which is labor-intensive and prone to error.

B. Food Quality and Traceability

Incorporating attention-based models helps in identifying visual cues associated with quality and defects, thereby:

- Improving the reliability of automated systems in identifying substandard or contaminated products.
- Supporting traceability and certification efforts in food logistics.

C. Advancement of Deep Learning Techniques

This project explores the integration of spatial and channel attention mechanisms (CBAM) and innovative loss functions (SIoU), contributing to the advancement of:

- **Model Interpretability:** Attention maps help in understanding the focus areas of the network.
- **Localization Accuracy:** SIoU enhances bounding box regression by accounting for alignment and skew.
- **Custom Architectures:** Demonstrates practical approaches to modifying base models for domain-specific challenges.

D. Open Research Directions

This work paves the way for further research in:

- Lightweight attention modules for edge devices.
- Adaptive loss functions tailored to specific object characteristics.
- Domain adaptation for agricultural and industrial datasets.

Overall, this project contributes to the broader effort of making intelligent visual systems more accurate, interpretable, and practically deployable in specialized sectors like agriculture and food processing.

VIII. Conclusion and Future Work

The integration of various architectural enhancements into the YOLO11vn model has led to measurable improvements in performance, though with some trade-offs. Below are the key findings:

- **Increased mAP**: The addition of **CBAM (Convolutional Block Attention Module)** and **C3k2 blocks** resulted in moderate to significant improvements in mAP (mean average precision), especially for cluttered and complex scenes. These modules help focus the model's attention on important regions of interest, improving accuracy for both small and large objects.
- **Depth and Complexity**: The model's depth increased to 364 layers, which significantly boosted the representational capacity. However, this also led to a higher computational load and an increase in **GFLOPs**, affecting training time and resource consumption. The increased depth might also be susceptible to overfitting if not managed with techniques like regularization or early stopping.
- **Improved Object Detection**: The combination of **SPPF** (Spatial Pyramid Pooling-Fast), **C2PSA** (Channel and Spatial Attention), and the upsampling + concat strategy further enhanced the multi-scale detection capability of the model, which is

crucial for identifying objects of varying sizes in the same scene. These modifications proved particularly beneficial for detecting overlapping objects.

- **Parameter Efficiency**: Although the model's parameter count rose to over 66 million, this increase resulted in better accuracy overall, especially when using high-resolution images. However, the trade-off in terms of training time and inference speed should be considered for deployment in real-time applications.

IX. Future Research Directions

While the current modifications show promising improvements, several future research directions can be pursued to optimize the model further:

- **Efficiency Optimization**:
  - **Pruning and Quantization**: Explore methods to reduce the number of parameters and computational complexity without compromising accuracy. This could involve pruning redundant connections or using lower precision for certain layers.
  - **Edge Deployment**: Investigate the deployment of the enhanced YOLO11vn model on edge devices like mobile phones or drones. Techniques like **knowledge distillation** could help retain performance while reducing model size for these constrained environments.
- **Attention Mechanisms**:
  - **Hybrid Attention Mechanisms**: The CBAM module has shown positive results, but future research could explore **transformer-based attention mechanisms** (e.g., Vision Transformers) for even better context modeling. Combining CBAM with self-attention layers could improve performance further, especially in highly variable environments.
  - **Dynamic Attention**: Implement dynamic attention strategies where the attention mechanisms adjust based on the difficulty of the task (e.g., focusing more on small objects in cluttered scenes and less on large objects).
- **Data Augmentation and Regularization**:
  - **Augmentation Techniques**: Future work could explore more advanced data augmentation techniques, especially in the context of rare or unseen object categories. Using **GANs** (Generative Adversarial Networks) to create synthetic data could further help in training the model for more robust generalization.
  - **Regularization**: With the increased model depth, regularization techniques like **DropBlock**, **DropPath**, or **Cutout** could be investigated to reduce overfitting and improve generalization.
- **Transfer Learning and Fine-Tuning**:
  - **Pretrained Models**: Investigating transfer learning with pretrained models from larger datasets like ImageNet or COCO could be beneficial. Fine-tuning on domain-specific data (e.g., medical images, satellite images) could improve performance for specific applications.
  - **Domain Adaptation**: If the model is used in specialized tasks (e.g., aerial imagery, autonomous driving), exploring domain adaptation techniques could help adapt the general model to new domains effectively.
- **Advanced Post-processing**:
  - **Non-Maximum Suppression (NMS)**: Further refine the NMS process to handle more overlapping objects effectively. **Soft-NMS** or **IoU loss-based NMS** could be explored to improve detection in challenging environments.
  - **Object Tracking**: In applications like video analysis or autonomous driving, integrating **object tracking** methods could help maintain continuity in object detection over time.
- **Real-time Inference**:
  - Given the increased model size and complexity, focusing on optimizing inference time for real-time applications is critical. **TensorRT**, **ONNX**, or hardware acceleration with **FPGA** or **TPU** support could drastically reduce inference time.

These future directions can lead to further advancements in YOLO-based models and contribute to a broader understanding of how different attention mechanisms and architectural modifications influence both accuracy and computational efficiency in deep learning models.

X. References

References

[1] A. Hayat, F. Morgado-Dias, T. Choudhury, T. P. Singh, and K. Kotecha, "Fruitvision: A deep learning based automatic fruit grading system," *Open Agriculture*, 2024. Received October 10, 2023; accepted February 27, 2024.

[2] A. K. Maitlo, R. A. Shaikh, and R. H. Arain, "A novel dataset of date fruit for inspection and classification," *Elsevier*, 2023. Institute of Computer Science, Shah Abdul Latif University Khairpur, Pakistan. Open access under CC BY license.

[3] M. Mohammed, N. K. Alqahtani, M. Munir, and M. A. Eltawil, "Applications of ai and iot for advancing date palm cultivation in saudi arabia," in *Internet of Things - New Insights*, IntechOpen, 2024.

[4] P. Rybacki, J. Niemann, S. Derouiche, S. Chetehouna, I. Boulaares, N. M. Seghir, J. Diatta, and A. Osuch, "Convolutional neural network (cnn) model for the classification of varieties of date palm fruits (phoenix dactylifera l.)," *Sensors 2024, 24, 558*, 2024.

[5] M. Ouhda, Z. Yousra, and B. Aksasse, "Smart harvesting decision system for date fruit based on fruit detection and maturity analysis using yolo and kmeans segmentation," *Journal of Computer Science*, vol. 19, no. 10, pp. 1242–1252, 2023.

[6] Q.-H. Phan, V.-T. Nguyen, C.-H. Lien, T.-P. Duong, M. T.-K. Hou, and N.-B. Le, "Classification of tomato fruit using yolov5 and convolutional neural network models," *Plants*, 2023.

[7] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

[8] G. Jocher, A. Chaurasia, L. Q, J. Fang, A. V, *et al.*, "Yolov8: Ultralytics open-source object detection architecture." https://github.com/ultralytics/ultralytics, 2023. Accessed: 2025-04-24.