Article

YOLOv8m for Automated Pepper Variety Identification: Improving Accuracy with Data Augmentation

Madalena de Oliveira Barbosa, Fernanda Pereira Leite Aguiar [ID], Suely dos Santos Sousa, Luana dos Santos Cordeiro, Irenilza de Alencar Nääs *[ID] and Marcelo Tsuguio Okano [ID]

Graduate Program in Production Engineering, Universidade Paulista, Rus Dr Bacelar 1212,
São Paulo 04026-002, SP, Brazil; madalena.divino@etec.sp.gov.br (M.d.O.B.);
fernanda.p.aaguiar@gmail.com (F.P.L.A.); suely.sousa2@etec.sp.gov.br (S.d.S.S.);
profluana.lourenco@gmail.com (L.d.S.C.); marcelo.okano1@docente.unip.br (M.T.O.)
* Correspondence: irenilza.naas@docente.unip.br

Featured Application

The study indicates the practical application of YOLOv8m, a deep learning architecture, for real-time identification and classification of pepper varieties, providing a practical solution for automating quality control and post-harvest sorting processes in food supply chains.

Abstract

This research addresses the critical need for an efficient and precise identification of *Capsicum* spp. fruit varieties within the post-harvest contexts to enhance quality control and ensure consumer satisfaction. Employing the YOLOv8m convolutional neural network, the study identified eight distinct pepper varieties: Pimento, Bode, Cambuci, Chilli, Fidalga, Habanero, Jalapeno, and Scotch Bonnet. A dataset comprising 1476 annotated images was utilized and significantly expanded through data augmentation techniques, including rotation, flipping, and contrast adjustments. Comparative analysis reveals that training with the augmented dataset yielded significant improvements across key performance indicators, particularly in box precision, recall, and mean average precision (mAP50 and mAP95), underscoring the effectiveness of data augmentation. These findings underscore the considerable potential of CNNs to advance the AgriFood sector through increased automation and efficiency. While acknowledging the constraints of a controlled image dataset, subsequent research should prioritize expanding the dataset and conducting real-world testing to confirm the model's robustness across various environmental factors. This study contributes to the field by illustrating the application of deep learning methodologies to enhance agricultural productivity and inform decision-making.

Keywords: pepper classification; YOLOv8m; computer vision in food supply; deep learning; object detection

1. Introduction

The international trade of spicy peppers (*Capsicum* species) constitutes a vibrant and economically significant sector within global agriculture and commerce [1]. Worldwide, the demand for spice peppers continues to rise, fueled by their role in ethnic cuisines, food preservation, and the growing recognition of their health-promoting properties, such as capsaicin content [1–3]. Major producing countries, including Brazil, India, China, and Mexico, utilize pepper cultivation as a vital agricultural activity that supports rural

livelihoods, exports, and local economies [1,4]. The stability of the pepper production chain ensures a constant market supply. The Food and Agriculture Organization (FAO) reported that global pepper production in 2022 reached 812,673 tons, with Brazil producing approximately 128,331 tons [4]. Peppers are essential spices and widely used horticultural crops with global cultivation origins in South America [1]. Peppers, commonly used in the culinary, pharmaceutical, and cosmetic industries, vary in color, heat, size, and shape. In Brazil, varieties such as Malagueta (*Capsicum frutescens*), Dedo-de-Moça (*Capsicum baccatum*), and Bode (*Capsicum chinense*) exhibit distinct traits, with seed variation influencing their characteristics [5].

Consumer satisfaction with spicy peppers is a multifaceted concept that significantly influences market demand and product differentiation. This satisfaction is not solely derived from heat level but also encompasses flavor profiles, aroma, freshness, and perceived health benefits [6,7]. Accurate identification of spicy varieties is crucial for optimizing production, enhancing quality, and minimizing errors in post-harvest management and market. Traditional post-harvest and market management screening methods are manual, time-consuming, and prone to errors in peppers (*Capsicum* spp.), which are widely cultivated and valued for their diverse flavors, shapes, and heat levels. These limitations underscore the need to adopt advanced technologies, such as computer vision, to streamline post-harvest practices [8,9].

The accurate identification of pepper (*Capsicum* spp.) varieties is a critical concern in agricultural production, as misclassification can result in substantial economic losses for producers, diminished product quality, and compromised supply chain efficiency. Given the increasing demand for specific pepper varieties in both domestic and international markets, failure to correctly identify and classify peppers can result in reduced market value, increased post-harvest losses, and a loss of consumer trust [1,4,5]. The complexity of distinguishing between morphologically similar varieties, particularly in large-scale operations, further worsens the inherent risks of errors associated with manual sorting and screening processes. Consequently, there is a need for advanced, automated identification methods to improve both the accuracy and efficiency of varietal discrimination in pepper production systems [6]. Recent developments in image processing have sparked significant interest in agricultural research [7]. The development and implementation of robust computer vision-based detection systems hold considerable potential for transforming post-harvest handling, quality control, and overall agricultural productivity by reducing human error, lowering operational costs, and ensuring consistent product standards [10,11].

Adopting computer vision methods provides greater accuracy and speed in the identification process. Recent advances in object detection methods, particularly those based on deep learning, have shown remarkable progress in agriculture, with several models producing positive results in identification tasks [12]. These technologies automate processes, reduce errors, enhance agricultural operations, and facilitate research advancements, as well as the integration of monitoring systems, contributing to more precise and sustainable agriculture. With the advancement of AI, deep learning has shown promise in addressing complex computer vision tasks in agriculture, with several deep learning algorithms, such as convolutional neural networks (CNNs), being successfully applied to recognition activities [13].

Integrating computer vision and artificial intelligence (AI), particularly with deep learning techniques, has revolutionized agricultural practices. Object detection models, such as convolutional neural networks (CNNs), have revealed substantial progress in automating tasks previously performed manually, enhancing speed and accuracy [12]. The YOLO (you only look once) model family stands out due to its real-time detection capabilities. However, image segmentation accuracy remains a critical factor for computer

vision applications, as it directly impacts the performance of identification models. Recent studies emphasize the importance of striking a balance between computational efficiency and segmentation fidelity, thereby ensuring robust results without compromising system performance [14].

Current research indicates that YOLO-based models are highly effective for detecting and classifying Capsicum (pepper) fruits, including tasks such as growth stage classification, segmentation, and real-time identification [10,15]. While YOLO and data augmentation are well established for general *Capsicum* detection and classification tasks [15,16], their application to efficiently and accurately identify pepper varieties remains underexplored.

The YOLOv8m (medium) model offers a favorable trade-off between detection accuracy and computational efficiency. It outperforms the smaller variants (nano and small) in precision while requiring significantly fewer resources than the larger models (large and extra-large), as previously verified in the current literature [16]. This outcome makes YOLOv8m a robust and efficient choice for deployment on hardware-constrained devices. The authors emphasize that the YOLO architecture was explicitly designed to be scalable, allowing researchers and practitioners to adjust model complexity according to the specific requirements of both the application and the available hardware. Moreover, YOLO-based models have been increasingly applied in the agriculture sector for tasks such as crop and pest detection, contributing to advancements in precision agriculture and process automation.

To address these challenges, the present study develops an automated method for identifying pepper varieties post-harvest using the YOLOv8m architecture and data augmentation. The main contributions include the construction of an annotated image dataset, the application of advanced deep-learning techniques, and a comprehensive evaluation of detection performance. The paper is structured as follows: the subsequent sections present the theoretical background, describe the materials and methods, report the results, and discuss the findings and implications for automated quality control in the post-harvest scenario.

2. Related Work

Computer vision is a branch of artificial intelligence that enables machines to interpret and analyze images to make decisions. This process utilizes cameras and computers to identify, track, and measure targets, followed by image processing [17]. This technology has increased automation and efficiency in various agricultural fields. Critical applications include crop growth monitoring, disease control, product identification and classification, automated harvesting, quality testing, and modern farm management automation. Unmanned aerial vehicles (UAVs) equipped with computer vision technology are used for aerial surveillance of agricultural lands, providing valuable data for decision-making. Automating identification processes through computer vision offers significant benefits, including increased efficiency and productivity, enhanced accuracy and consistency, early problem detection, cost reduction, and continuous monitoring [17].

Deep learning (DL), a subfield of machine learning (ML) and artificial intelligence (AI), is considered a core technology of the Fourth Industrial Revolution, also known as Industry 4.0 [18,19]. The word "deep" refers to the multiple layers through which data is processed to build a hierarchical data-driven model. DL models can automatically extract features from data, enabling them to handle complex tasks accurately. These models are widely applied in healthcare, cybersecurity, business intelligence, and especially agricultural visual recognition tasks.

According to Li et al. [20], convolutional neural networks (CNNs) represent a fundamental class of deep learning models, particularly significant in the field of computer vision.

The architecture of CNNs is designed to process grid-structured data, such as images. This process is achieved through convolutional layers, which apply convolution operations to extract local features from images, followed by pooling layers that reduce the dimensionality of the features while retaining essential representations. These networks have shown remarkable capabilities in learning complex hierarchical representations directly from data, making them highly effective in pattern recognition tasks such as image classification, object detection, and semantic segmentation. The capacity of CNNs to capture detailed and abstract visual patterns makes them ideal for identifying various visual characteristics of agricultural products, such as shape, color, and texture, which are critical for distinguishing between different pepper varieties [21].

Recent studies show that deep learning methods substantially enhance the detection and classification of agricultural diseases and renewable energy sites. Ioannou and Myronidis [22] used convolutional neural networks (CNNs) with high-resolution satellite imagery to automatically and accurately detect photovoltaic farms, offering a scalable tool for energy monitoring. In agriculture, Arnob et al. [23] found that ResNet50 outperformed other deep-learning models in classifying cauliflower diseases, achieving an accuracy of 90.85%. Roy and Kukreja [24] applied vision transformers (ViTs) for rice leaf disease detection and severity estimation, surpassing some limitations of traditional CNNs and achieving strong performance in both classification and severity assessment. Together, these findings highlight the value of advanced AI models in both environmental monitoring and precision agriculture.

Hussain [25] explains that the YOLO (you only look once) model represents an innovative approach to real-time object detection, using a single convolutional neural network that simultaneously predicts bounding boxes and class probabilities from an image divided into a grid. This architecture enables the model to process the entire image simultaneously, resulting in high frame-per-second rates suitable for critical applications such as surveillance, autonomous driving, and industrial automation. In post-harvest agricultural contexts, YOLO's efficiency is emphasized by its ability to perform accurate real-time detections, supporting real-time monitoring and identification essential for dynamic agricultural environments, such as fields or conveyor systems sorting produce [26].

The YOLO architecture has evolved substantially since its inception, with each version introducing significant enhancements to improve object detection accuracy, speed, and computational efficiency. YOLOv1 marked the initial step toward real-time detection using a grid-based approach to predict bounding boxes. Building on this, YOLOv2 incorporated anchor boxes, significantly improving localization accuracy. YOLOv3 further refined the model by adopting the Darknet-53 architecture and integrating feature pyramid networks (FPNs), enabling better generalization across object scales. In YOLOv4, additional architectural innovations were introduced, including the Bag of Specials (BoS), which improved accuracy without sacrificing speed. A significant shift occurred with YOLOv5, implemented in PyTorch, where optimizations in training processes and the integration of Cross Stage Partial Network (CSPNet) and Path Aggregation Network (PANet) enhanced performance and modularity. YOLOv6 introduced reparameterization techniques and novel backbone designs for greater efficiency. Continuing the trend of architectural improvements, YOLOv7 employed Efficient Layer Aggregation Network (ELAN) blocks to improve feature learning. With YOLOv8m, the focus turned to computational optimization and more refined training strategies, making it suitable for resource-constrained environments. YOLOv9 introduced programmed gradient information (PGI) to further enhance detection precision and efficiency. Most recently, YOLOv10 eliminated the traditional non-maximum suppression (NMS) step, streamlining the detection of multiple overlapping objects and enhancing throughput for dense scenes. This progression reflects a continuous drive to balance speed,

accuracy, and model simplicity, particularly in real-time agricultural applications such as pepper variety identification [11].

These developments position YOLO as a leading architecture for real-time visual classification systems in the agricultural sector. In parallel, other deep learning approaches, such as convolutional neural networks (CNNs) combined with transfer learning, have also indicated strong practical potential in the agricultural sector. Previous studies have shown their effectiveness, particularly in the pepper production chain. Ren et al. [27] developed a model to classify bell peppers (*Capsicum annuum*) based on external quality, utilizing pre-trained neural networks and image capture tools. Their approach achieved high accuracy (98.14%) and low loss (0.0669), showing excellent performance and generalization without requiring manual feature extraction.

Kurtulmuş et al. [28] investigated the classification of pepper seeds using computer vision and neural networks. Their study involved extracting color, shape, and texture features from the seeds and using a multilayer perceptron (MLP) for classification, achieving an accuracy of 84.94%. This advancement underscores the feasibility of applying CNNs at various stages of the pepper production chain, from seed to final product. Furthermore, a deep learning model utilizing convolutional neural networks (CNNs) for disease detection in pepper leaves was developed based on a dataset of 2478 images.

Previous research [29] has emphasized the use of transfer learning to enhance diagnostic accuracy under cultivation conditions. Their results presented an accuracy of 99.55% in distinguishing between healthy and diseased leaves, highlighting the model's practical applicability for early disease detection in agriculture.

Presenting additional application lines, Mohi-Alden et al. [30] developed an intelligent classification system for bell peppers using deep convolutional neural networks (DCNN). By employing a modified ResNet50 architecture, the system achieved an accuracy of 96.89% in classifying bell peppers based on maturity stage and size. The system was integrated into a computer vision-based sorting machine, showing significant potential to improve classification processes in the food industry, with an overall sorting rate of approximately 3000 samples per hour.

Research on applying CNNs and transfer learning to classify and detect quality, types, and diseases in bell peppers reveals significant potential to transform food quality assessment practices. Previous studies have proven that integrating these technologies can enhance the accuracy, reduce costs, and increase the efficiency of agricultural processes [31].
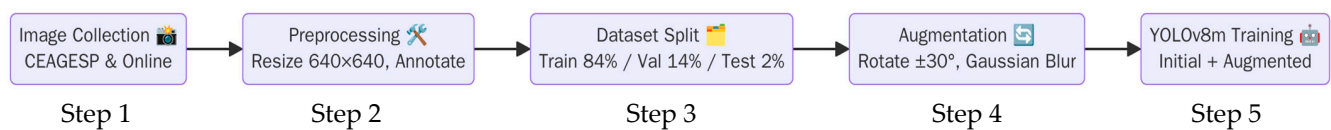
3. Materials and Methods

3.1. Data Recording

Figure 1 presents a schematic overview of the five-step methodology employed for automated pepper variety identification using the YOLOv8m model. Step 1 involved image collection, where photographic images of eight *Capsicum* fruit varieties were acquired to establish the initial dataset. Step 2 involved image annotation, where the collected images were standardized to a 640 × 640 pixel resolution and annotated with bounding boxes to identify *Capsicum* fruit instances using the Roboflow platform. In Step 3, we performed the initial model training using the YOLOv8m model. Step 4 involved data augmentation, where we enhanced the model's generalization and expanded the dataset to include larger images through data augmentation techniques. In Step 5, the YOLOv8m model underwent final training using the full and augmented dataset.

The first step involved training the YOLOv8m model using a dataset comprising 1476 images representing eight distinct pepper varieties. Initially, the dataset consisted of approximately 300 images, collected exclusively through in-person visits to the Company of Warehouses and General Stores of São Paulo (CEAGESP (a major public enterprise

in Brazil and one of the largest wholesale food centers in the world and the largest in Latin America). Located in São Paulo, Brazil, CEAGESP plays a crucial role in the supply chain of horticultural products, fish, flowers, and other food items. The main unit in São Paulo is the largest such center in South America and handles a significant volume of food products annually, where photographs of the eight pepper types were captured under real market conditions using various devices, including the Samsung Galaxy Tab A8 tablet, which features an 8 MP rear camera; the Samsung Nikon Coolpix L120, a compact camera with a 14.1 MP CCD sensor and a 21 × optical zoom lens; and the iPhone 11 smartphone, which has a dual 12 MP camera and 4 K video recording capability. These devices ensured high-quality, versatile image capture in diverse market settings. However, as the research progressed, it became evident that expanding the number of pepper varieties was essential to enhance the model's generalization capacity and practical applicability [32].

Image Collection 📷 CEAGESP & Online	Preprocessing 🔧 Resize 640×640, Annotate	Dataset Split 📑 Train 84% / Val 14% / Test 2%	Augmentation 🔄 Rotate ±30°, Gaussian Blur	YOLOv8m Training 🤖 Initial + Augmented
Step 1	Step 2	Step 3	Step 4	Step 5

Figure 1. Steps adopted in the current research to identify *Capsicum* fruits.

Another critical factor was the need to incorporate greater product variability. Observations at CEAGESP indicated that the peppers available for commercial sale exhibited high-quality standards, resulting in limited variation in size, coloration, and overall product condition. This high level of visual uniformity could potentially restrict the model's ability to accurately identify peppers in more diverse real-market conditions, where variations in texture, ripeness, lighting, and shape are more common. An alternative approach was adopted to expand the dataset to address this limitation: sourcing images from online platforms, including open-access databases such as Google Images under the Creative Commons license. Image selection was based on quality criteria, prioritizing high resolution ($\geq$300 dpi) and a diversity of angles to ensure a dataset that is more representative of the commercial reality of peppers across different contexts.

In the second step, the images were normalized to 640 px × 640 px and annotated with bounding box annotations using the Roboflow platform [33]. As Matuck et al. [34] described, Roboflow is a computer vision tool that streamlines the training and deployment of real-time object detection models.

The platform supports image importation, labeling, data preprocessing, model training, and deployment, making the development of computer vision models more accessible and efficient. Through this platform, the images were annotated, and distinct classes were created for each pepper variety: Biquinho Pepper, Bode Pepper, Cambuci Pepper, Chilli Pepper, Fidalga Pepper, Habanero Pepper, Jalapeno Pepper, and Scotch Bonnet Pepper.

Table 1 presents a photographic reference and concise morphological description for the eight pepper cultivars contained in the study (Biquinho, Bode, Cambuci, Chilli, Fidalga, Habanero, Jalapeno, and Scotch Bonnet). Each cultivar is provided with its common name, scientific designation, and key visual traits (shape, color, color variation, and size). These descriptors justify the class labels employed during image annotation and highlight the fine-grained features the YOLOv8m network must learn to distinguish during automated detection.

Table 1. Image, name, variety, and morphological description of the eighth studied pepper.

Image of the Peppers	Name/Variety	Description
	Biquinho/*Capsicum chinense*	Small, teardrop-shaped, glossy red fruits measuring 2–3 cm long and 1–2 cm in diameter.
	Bode/*Capsicum chinense*	Round and bulbous, yellowish, typically yellow or red, 2–4 cm in both height and diameter, with a distinctly wrinkled surface.
	Cambuci/*Capsicum baccatum*	Bell-shaped with pronounced lobes, green in color, measuring 4–6 cm in diameter and 3–5 cm in height.
	Chilli/*Capsicum annuum*	Elongated and slender, bright red or chocolate, typically 8–12 cm long and 1–2 cm in diameter.
	Fidalga/*Capsicum chinense*	Small and round, shiny yellow, about 2–3 cm in diameter and 2–4 cm in height.
	Habanero/*Capsicum chinense*	Oval and lantern-shaped, orange, 3–6 cm long and 2–4 cm in diameter, with a wrinkled texture.
	Jalapeno/*Capsicum annuum*	Medium-sized, cylindrical, dark green, measuring 5–9 cm in length and 2–4 cm in diameter, with a tapered end.
	Scotch Bonnet/*Capsicum chinense*	Squashed-round shape, yellow, 4–6 cm in both height and diameter, with a wrinkled appearance.

3.2. Dataset Partitioning and Preprocessing Strategies

In the third step, the dataset was then divided into three distinct subsets to increase the volume and diversity of training samples, considering the limitations of the initial dataset:

- Training Set: 1245 images (84% of the total)
- Validation Set: 201 images (14% of the total)
- Test Set: 30 images (2% of the total)

Although the final test set comprised only 30 images (1% of the total dataset), this decision prioritized maximizing the volume and diversity of training data, which is critical for enhancing deep learning performance in object detection tasks. Although the test set was relatively small (30 images), it was carefully constructed to ensure visual diversity, encompassing variations in lighting, angles, and fruit morphology, thereby reflecting real-world application scenarios. To reduce the risk of overfitting, data augmentation techniques such as rotation and Gaussian blur were employed to introduce controlled variability and improve the model's generalization capacity. The stabilization of performance metrics after 100 epochs, along with consistent improvements observed with the use of augmented data, indicates a well-adjusted model. Approximately 900 instances were generated for each pepper variety at the end of the process. Subsequently, a second round of model training was conducted using an augmented dataset.

This methodological approach is consistent with the findings of Frizzi et al. [35], who verified that in data-scarce scenarios, a larger training partition combined with data augmentation effectively mitigates overfitting, provided the validation and test sets retain

representative variability. While this principle guides our methodology, we acknowledge that the limited size of the current test set constrains a definitive evaluation of the risk of overfitting. Future research should incorporate strategies such as cross-validation and an expanded test dataset to enhance the robustness and reliability of the findings.

3.3. Data Augmentation

In the fourth step, we employed data augmentation. This process enlarges a training dataset through techniques such as rotation, flipping, cropping, and contrast adjustment, thereby enhancing the model's generalization and performance [35]. Data augmentation techniques were applied to the training set to improve the generalization capacity of the YOLOv8m model and optimize its accuracy in pepper identification. This strategy aimed to simulate natural variations in real-world image capture conditions, making the model more robust to changes in lighting, angles, and image sharpness. The augmentation methods, such as rotation and blurring, were selected based on prior studies indicating their positive impact on YOLOv8m accuracy. In contrast, cropping was excluded due to its tendency to eliminate critical features in small objects [36].

In this study, data augmentation techniques were applied strategically in two distinct phases. Initially, the YOLOv8m model was trained on the original dataset to establish a baseline for performance. Subsequently, data augmentation methods such as rotation and Gaussian blur were introduced during a second training cycle, allowing for a more controlled assessment of their specific impact on the model's generalization capabilities. Although it is common practice to apply data augmentation before the initial training phase, as noted by Ioannou and Myronidis [22], the adopted approach offered clearer insights into how augmentation influenced convergence behavior and class-level detection performance. The observed improvements in metrics such as box precision and mAP during the second phase demonstrated that post-baseline augmentation not only enhanced generalization but also contributed to the model's robustness in recognizing subtle morphological differences among pepper varieties. Additionally, this phased methodology enabled more efficient use of computational resources by avoiding unnecessary dataset expansion before validating the model's initial learning performance.

Based on these indications, the following procedures were applied to the training dataset:

- Rotation: Images were randomly rotated at angles between $-30°$ and $+30°$, increasing perspective variability and reducing the model's dependence on fixed angles.
- Blurring: A soft Gaussian blur filter (3×3 kernel) was applied to simulate focus variations that may arise due to camera movement or differences in image quality.
- Following data augmentation, the dataset increased to a total of 3964 images, with the final data split structured as follows:
- Training Set: 3733 images (94% of the total)
- Validation Set: 201 images (5% of the total)
- Test Set 30 images (1% of the total)

The dataset was divided into training (94%), validation (5%), and test (1%) subsets to increase the volume and diversity of training samples, considering the limitations of the initial dataset. Tables 2 and 3 provide the specifications necessary for reproducibility of the training process. This configuration represents a standard cloud-based training setup, ensuring that the model can be reproduced under similar computational conditions.

Hyperparameters were defined based on the default configurations provided in the defaults.yaml file of the official Ultralytics YOLOv8 implementation. These values reflect established best practices for training object detection models [37]. The use of standardized configurations ensures methodological consistency with the YOLOv8 framework and facilitates the comparability of results across related studies.

Table 2. Hardware specifications and relevant software environments for the reproducibility of the training process.

Parameter	IT Environment
Operating system	Ubuntu 20.04/Google Colab VM
Processor	Intel(R) Xeon(R) CPU @ 2.20 GHz
RAM	13 GB
GPU	NVIDIA Tesla T4 (16 GB VRAM)
CUDA version	11.8
Python version	3.10
YOLOv8 version	Ultralytics 8.1.0
PyTorch version	2.0.1
Training environment	Google Colab

Table 3. Key hyperparameters adopted for training the YOLOv8m model.

| Hyperparameter	Value	Description
epochs	200	Number of training iterations
batch_size	16	Training samples per batch
imgsz	640	Input image resolution (square)
lr0	0.01	Initial learning rate
lrf	0.01	Final learning rate decay factor
momentum	0.937	SGD momentum
weight_decay	0.0005	Regularization parameter
optimizer	auto	Optimizer automatically selected (typically SGD)
patience	100	Early stopping patience (epochs without gain)
mosaic	1.0	Mosaic augmentation enabled
fliplr	0.5	Horizontal flip probability
translate	0.1	Image shift augmentation
scale	0.5	Image scaling range for augmentation

3.4. YOLO Usage

The selection of the YOLOv8m model is justified by its superior performance in agricultural contexts, particularly in terms of accuracy, inference speed, and model compactness, compared to alternatives such as EfficientDet and CenterNet. YOLO's architecture, which frames object detection as a single-pass regression problem, provides a significant velocity advantage crucial for real-time applications [12].

This claim is substantiated by a systematic review of 30 studies by Badgujar et al. [26], which concluded that YOLO models consistently outperform frameworks such as SSD, Faster R-CNN, and EfficientDet in agricultural tasks. Given its verified efficiency and accuracy, YOLOv8m is well suited for the present study on varietal pepper classification in uncontrolled environments. Figure 2 illustrates the flow of features through convolutional layers, C2f modules, and the anchor-free detection head.

The fifth step involved model training conducted using the Google Colab Pro infrastructure [38], which ensured access to hardware acceleration for deep learning tasks. The computational configuration comprised an NVIDIA Tesla T4 GPU with 15 GB of VRAM, driver version 550.54.15, and CUDA version 12.4. Available memory was 15.36 GB, with GPU power consumption ranging from 9 W (minimum usage) to 70 W (maximum capacity).

The frameworks used were Python 3.10.12 [39], PyTorch 2.5.1 [40], and Ultralytics YOLOv8m [37], along with OpenCV [41] and Roboflow [33]. Google Colab Pro was selected due to its access to high-performance GPUs, which enables the efficient training of the YOLOv8m model within a reduced time frame. The programming language used was Python [39], in combination with several specialized libraries, including:

- OpenCV: For image manipulation.
- TensorFlow: For model construction and training.
- PyTorch: An alternative deep learning framework [40].
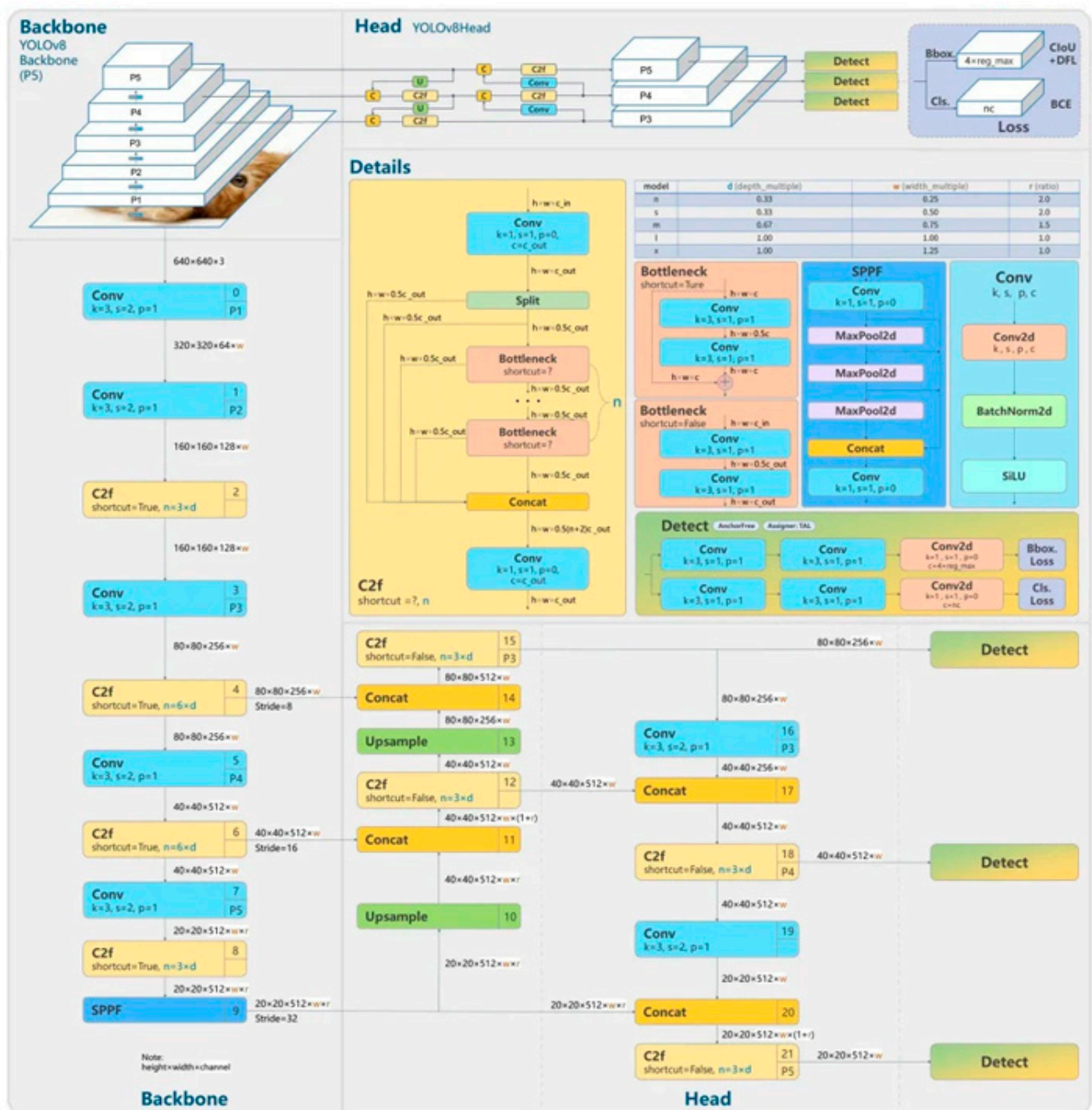
Figure 2. Diagram of the YOLOv8 architecture, including backbone, neck, and head components. Adapted from [33].

After training, the model's performance was evaluated using the following metrics provided by the YOLOv8m CNN (Equations (1)–(3)):

- Box Precision (BoxP): Measures the accuracy of bounding box predictions relative to the annotated boxes (Equation (1)).

$$\text{Precision} = TP/(TP + FP) \tag{1}$$

where TP = true positive, and FP = false positive. The TP is a predicted bounding box that correctly matches a ground truth bounding box of the same class. FP is a predicted bounding box that does not correspond to a real object.

- Recall (R): Measures the proportion of true positives among all positive instances in the dataset (Equation (2)).

$$\text{Recall} = TP/(TP) + (FN) \tag{2}$$

where TP = true positive, and FN = false negative. The FN is a ground truth bounding box that the model failed to detect.

- Mean Average Precision at 50% IoU (mAP50): Calculates the average precision across different intersection-over-union (IoU) thresholds, focusing on a 50% threshold (Equation (3)). The mAP50 is a standard metric for evaluating object detection models. It represents the average of the average precision (AP) calculated for each object class. For this metric, a predicted bounding box is considered a true positive only if its intersection over union (IoU) with a ground truth box of the same class is 0.50 or greater. The AP for each class is the area under its corresponding precision–recall curve, reflecting the trade-off between precision and recall at varying detection confidence thresholds under the 0.50 IoU criterion.

$$mAP_{50} = (1/N) \times \Sigma [AP_{c, IoU = 0.50}] \text{ from } c = 1 \text{ to } N \tag{3}$$

where N = total number of object classes, and AP_c, 0.50 = average precision for class c, calculated using an IoU threshold of 0.50. This metric refers to the area under the precision–recall curve for class c, where predictions are counted as true positives if their IoU with ground truth is greater than 0.50.

- Mean average precision at 95% IoU (mAP95): Calculates the average precision at a stricter 95% IoU threshold (Equation (4)).

$$mAP_{95} = (1/N) \times \Sigma [AP_{c, IoU = 0.95}] \text{ from } c = 1 \text{ to } N \tag{4}$$

where N = total number of object classes, and AP_c, 0.95 = average precision for class c, calculated using a fixed IoU threshold of 0.95. This metric is the area under the precision–recall curve for class c, where predictions are counted as true positives if their IoU with ground truth is greater than 0.95.

The YOLOv8m model was chosen for the present study due to its proven efficiency in real-time object detection tasks, aligning with the precision and speed requirements for identifying pepper varieties. This version represents a significant advancement within the YOLO family, combining architectural enhancements and advanced training techniques, making it particularly suited for agricultural applications. According to Terven and Cordova-Esparza [11], YOLOv8m achieved superior performance in metrics such as mean average precision (mAP) compared to earlier models, including YOLOv5, and even competing approaches like Faster R-CNN. The results were compared between datasets with and without augmentation, highlighting the impact of data augmentation on the YOLOv8m model's performance in pepper detection. Figure 3 shows the workflow of the methodology.
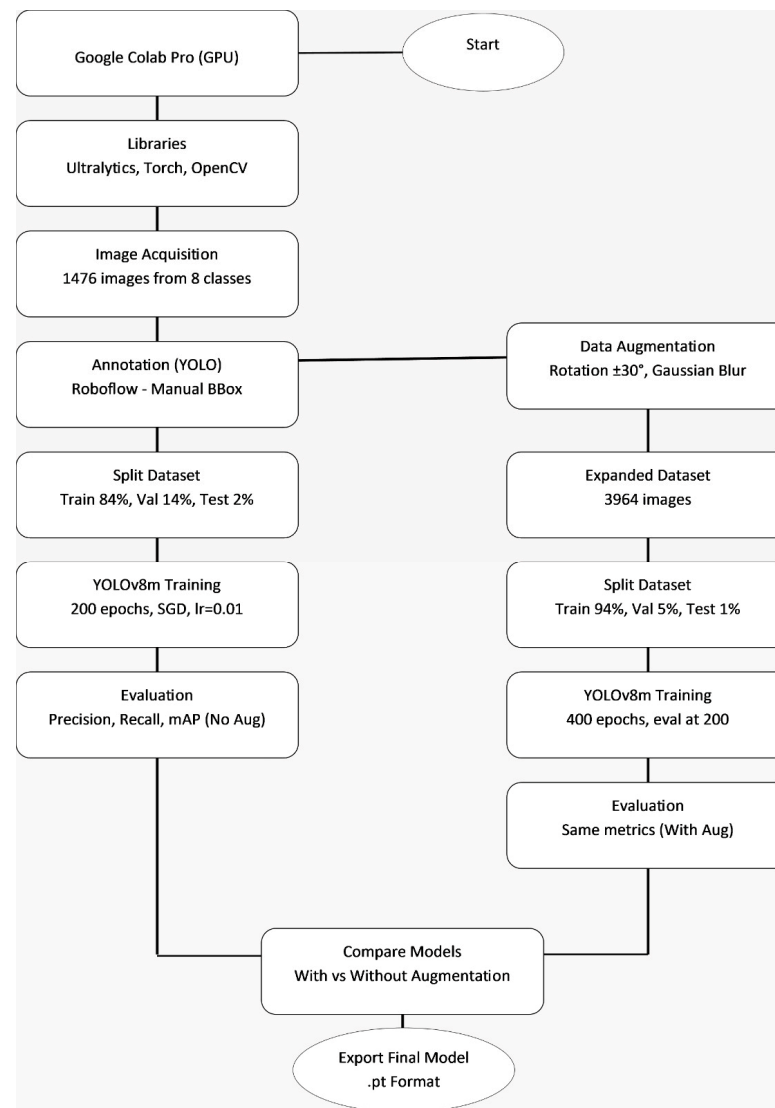
Figure 3. Methodology workflow.

4. Results

This study investigated the training of the YOLOv8m convolutional neural network (CNN) to identify eight distinct varieties of peppers. The model's performance was evaluated both with and without the application of data augmentation techniques, considering key metrics such as bounding box precision (BoxP), recall, and mean average precision across different intersection over union (IoU) thresholds (mAP50-95).

Table 4 summarizes the evolution of global detection metrics—bounding-box precision (BoxP), recall, mAP$_{50}$, and mAP$_{50-95}$—at four representative training epochs (10, 50, 100, 170) for models trained with and without augmentation. The results indicate that rotation and Gaussian-blur augmentation accelerated convergence and yielded substantial gains.

Table 4. Summary of the model training to identify peppers.

Epochs	BoxP	BoxP with Aug.	Recall	Recall with Aug.	mAP50	mAP50 with Aug.	mAP50-95	mAP50-95 with Aug.
10	0.169	0.340	0.206	0.387	0.160	0.360	0.061	0.163
50	0.526	0.705	0.570	0.605	0.562	0.696	0.278	0.369
100	0.626	0.749	0.599	0.576	0.641	0.676	0.334	0.403
170	0.694	0.684	0.600	0.638	0.673	0.694	0.361	0.371

The results presented in Table 4 indicate the model's performance progression throughout the training process, highlighting the positive effects of data augmentation. The application of these techniques led to improvements across all evaluated metrics.

Continuing training beyond 100 epochs revealed the stabilization of performance metrics, indicating that the model had reached an appropriate point of convergence for the dataset. This behavior suggests that increasing the number of epochs beyond this threshold would not yield significant improvements in the model's learning performance.

Figure 4 shows that the results display consistent improvements in box precision, recall, and mean average precision (mAP) when augmentation techniques were applied, indicating enhanced generalization capabilities of the model.
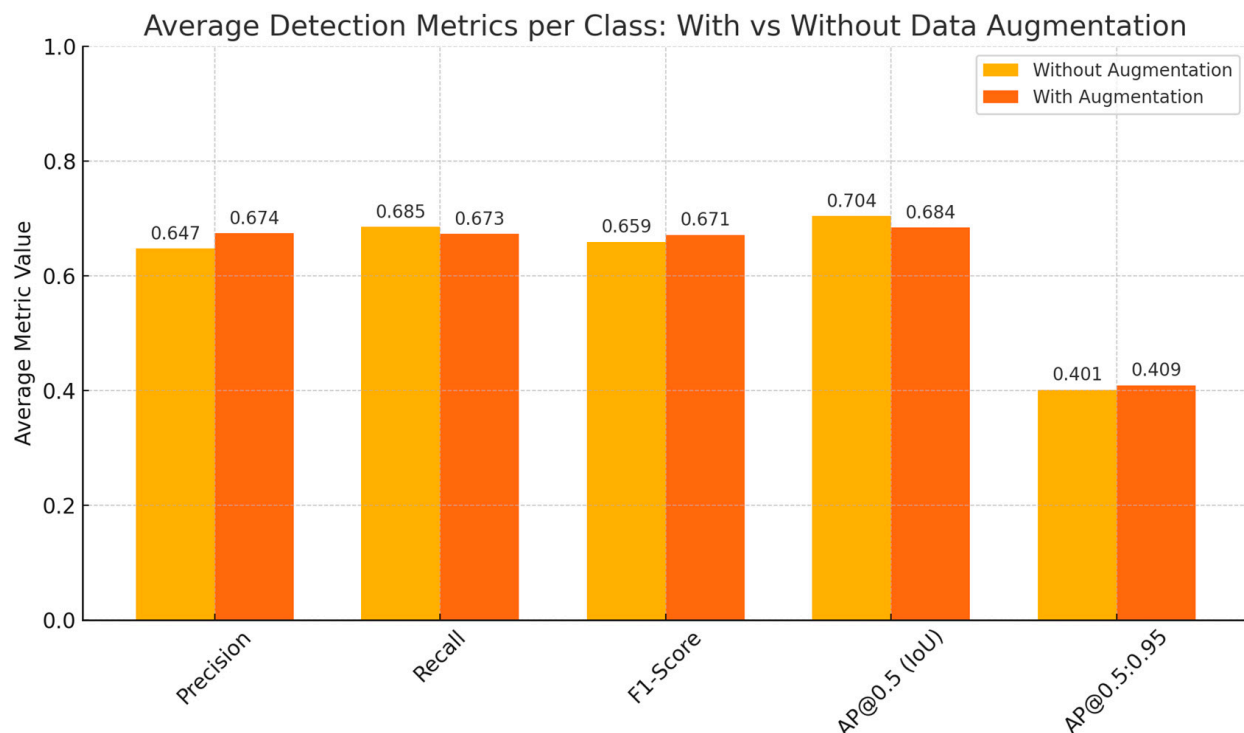
Figure 4. Comparison of object detection performance metrics between training with and without data augmentation.

The comparative analysis between models trained with and without data augmentation techniques revealed consistent improvements in the average values of precision, F1-score, and AP@0.5:0.95. These results underscore the importance of incorporating data augmentation methods to enhance the robustness and generalization capacity of object detection models. As emphasized by Abdulkareem et al. [42], applying geometric transformations such as rotation, flipping, and scaling increases the visual diversity of the training dataset. This broader variability enables convolutional neural networks, particularly those based on the YOLO architecture, to detect objects more accurately across different orientations, scales, and visual perspectives.

Table 5 provides the class performance values (BoxP, recall, mAP_{50}, and mAP_{50-95}) for the best non-augmented and augmented models after 170 epochs. The table highlights how varietal morphology affects detector sensitivity and where further targeted data enrichment may be necessary.

Table 5. Result of each class in the last training.

Class	BoxP	BoxP with Aug.	Recall	Recall with Aug.	mAP50	mAP50 with Aug.	mAP50-95	mAP50-95 with Aug.
Biquinho-Pepper	0.693	0.658	0.516	0.486	0.636	0.562	0.692	0.386
Bode-Pepper	0.618	0.638	0.436	0.564	0.523	0.597	0.566	0.293
Cambuci-Pepper	0.747	0.852	0.758	0.777	0.816	0.827	0.884	0.596
Chilli-Pepper	0.713	0.750	0.595	0.729	0.648	0.790	0.622	0.345
Fidalga-Pepper	0.768	0.652	0.579	0.592	0.701	0.679	0.626	0.377
Habanero-Pepper	0.636	0.577	0.613	0.667	0.641	0.693	0.748	0.675
Jalapeno-Pepper	0.622	0.595	0.775	0.75	0.702	0.721	0.422	0.489
ScotchBonnet-Pepper	0.756	0.746	0.527	0.542	0.720	0.708	0.404	0.415

The data show that the impact of data augmentation varied across classes, with some varieties benefiting more from the technique than others (Supplementary material). Peppers that showed improvement with data augmentation:

- Cambuci-Pepper achieved the highest gains through data augmentation, reaching 0.852 in BoxP (+14%) and 0.827 in mAP50 (+1.1%).
- Chilli-Pepper also identified significant improvements, particularly in recall (+22%) and mAP50-95 (+31%).

The peppers with more stable or decreased performance with data augmentation were the Fidalga Pepper, which showed a reduction in precision (BoxP dropped from 0.768 to 0.652), suggesting that the model may have been affected by noise in the newly generated samples, and Habanero-Pepper, which also showed a decline in BoxP (from 0.636 to 0.577); however, it showed an improvement in recall (+8.8%), indicating that more objects were detected, even though with lower precision.

The peppers that consistently performed well under both conditions were the Jalapeno pepper and Scotch Bonnet pepper, which exhibited minimal variation between tests. This result suggests that the model effectively captures their visual characteristics, regardless of the training strategy.

Observations of the results on test images also raise several considerations. Figure 5 presents eight subfigures (a to h) showing the YOLOv8m model's detection of each pepper class on test images. The eight pepper varieties evaluated in the study are (a) Biquinho, (b) Bode, (c) Cambuci, (d) Chilli, (e) Fidalga, (f) Habanero, (g) Jalapeno, and (h) Scotch Bonnet. The bounding boxes inside the images highlight the detections made by the model, along with the confidence scores assigned to each prediction.

The confidence values vary from image to image, as they depend on the specific conditions of each sample evaluated by the model. This variation in confidence arises from multiple factors, including lighting and contrast, the position and angle of the peppers, instance overlap, image quality, and resolution. Such fluctuations in confidence values are expected and reflect the model's robustness when dealing with diverse capture conditions. However, the variability in prediction confidence suggests that the model can still be optimized to reduce fluctuations in values across different images. Strategies such as adjusting the detection threshold, refining class balance, and implementing post-processing techniques could help improve the consistency of predictions.
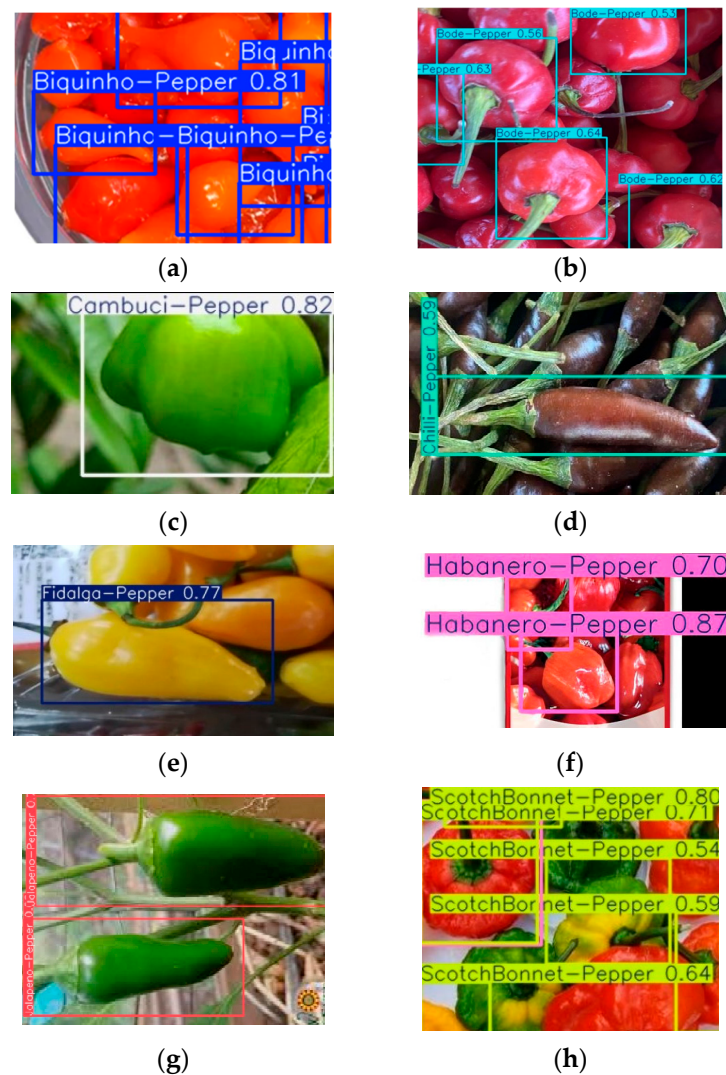
Figure 5. Example of results of the model tests using samples of the *Capsicum* fruits: (**a**) Biquinho; (**b**) Bode; (**c**) Cambuci; (**d**) Chilli; (**e**) Fidalga; (**f**) Habanero; (**g**) Jalapeno; and (**h**) Scotch Bonnet. The bounding box defines the spatial location and extent of the detected object.

5. Discussion

The reduction in precision observed for the Fidalga-Pepper and Habanero-Pepper classes may be attributed to the introduction of variations that do not accurately represent the actual distribution of the training images. The reduced performance observed in certain pepper varieties, such as Fidalga, is likely attributable to their visual similarity to other yellow, rounded types like Habanero and Scotch Bonnet, combined with the possible introduction of unrealistic variations during data augmentation, which may have impaired the model's ability to distinguish subtle inter-class differences. This effect has already been documented in the literature, as noted by Yilmaz and Kutbay [36], who indicate that data augmentation techniques can, in some cases, generate unrealistic instances, hindering the model's generalization capacity. The variation in performance across classes suggests that some pepper varieties possess visual characteristics that facilitate detection, while others require further refinement of the model to improve detection.

The application of data augmentation also impacted the total training time of the model. Without augmentation, training was completed in 200 epochs, totaling 3 h and 43 min. Using the augmentation technique, there was a reduction in epochs (170), but the processing time more than doubled, reaching 6 h and 54 min. This increase in processing

time, from 3 h and 43 min without augmentation to 6 h and 54 min with augmentation, can be attributed to the larger volume of images and the increased computational complexity introduced by greater sample diversity. This impact on training time aligns with the findings of Yilmaz and Kutbay [36], who report that applying data augmentation techniques can significantly increase model execution time, especially when implemented on large datasets. The increase in training time needs to be weighed against the gains in model accuracy. Although data augmentation yielded better performance for some pepper classes, the computational cost remains a critical factor for deploying the model in real-world applications.

The present study's findings confirm previous research [43], indicating the effectiveness of computer vision in agricultural product identification while highlighting the challenges associated with automated pepper detection. An earlier study [17] highlights the increasing application of computer vision in agricultural automation, noting that CNN-based models can replace traditional screening methods, thereby reducing time and minimizing human error. Similarly, Li et al. [10] show that the performance of CNNs is highly dependent on the quantity and diversity of the dataset, which explains the positive impact of data augmentation observed in the present study.

The choice of YOLOv8m for this study aligns with research by Hussain [25] and Terven and Cordova-Esparza [11], who highlight the YOLO architecture's advances in efficiency and speed in agricultural object detection. This study reinforces these findings, demonstrating that YOLOv8m can accurately identify pepper varieties, especially when trained on a diverse dataset.

We observed a clear trend based on the experimental results: model performance, as indicated by metrics such as mAP50 and AP50-95, generally improves with increased training from 10 to 200 epochs without data augmentation. However, further training to 400 epochs yields diminishing returns and introduces the risk of overfitting. In contrast, data augmentation, particularly evident after 50 epochs, substantially enhances the model's generalization capabilities across the eight distinct pepper varieties. This result points out the crucial role of data augmentation in enabling extended training durations to achieve robust and accurate fine-grained classification.

The YOLOv8m model indicated a notable improvement in performance after applying data augmentation techniques. Key metrics such as box precision (BoxP) and recall showed significant gains after 50 epochs, with further stabilization observed after 100 epochs. For instance, the mAP50 for Chilli-Pepper increased from 0.648 to 0.790 with augmentation, while the recall for Bode peppers improved from 0.436 to 0.564. The precision and recall values across the eight pepper varieties indicate the model's capacity to differentiate effectively between them. These results underscore the crucial role of data augmentation in enhancing the performance of CNN models in agricultural contexts, laying a solid foundation for future advancements in automated crop identification and management systems.

Future research should explore advanced architectural enhancements, such as integrating attention mechanisms (e.g., SE-Net, CBAM), to improve spatial and channel feature refinement. Alternatively, employing ensemble approaches that combine predictions from multiple models could increase robustness across heterogeneous field conditions. Investigating lightweight architectures (e.g., MobileNet, YOLO-NAS) may also support deployment in edge devices used in precision agriculture.

6. Conclusions

The present study presents a practical application of the YOLOv8m CNN in identifying eight distinct *Capsicum* pepper varieties. Implementing data augmentation techniques significantly enhanced the model's precision, recall, and mean average precision (mAP)

metrics. These findings underscore the potential of deep learning to revolutionize food quality practices, particularly in automating management tasks.

Although this research identified the effectiveness of the YOLOv8m model for pepper identification, further research is essential to enhance the model's robustness in real-world scenarios by including images captured under diverse real-world conditions, such as varying lighting intensities, partial occlusions from overlapping produce, heterogeneous backgrounds, and differences in image resolution and camera angles. Exploring advanced techniques, such as transfer learning and alternative object detection models, could improve the model's generalization capacity and adaptability.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You only look once (real-time object detection framework)
CNN	Convolutional neural network
MobileNet	CNN architecture developed by Google, designed for mobile applications
SE-Net	Squeeze-and-excitation network (a channel attention mechanism for improving CNN performance)
CBAM	Convolutional block attention module (a lightweight attention module combining spatial and channel attention)
YOLO-NAS	YOLO with neural architecture search (an optimized variant of YOLO using automated architecture search for improved accuracy and efficiency)

References

1. Riaz, U.; Iqbal, S.; Sohail, M.I.; Samreen, T.; Ashraf, M.; Akmal, F.; Siddiqui, A.; Ahmad, I.; Naveed, M.; Khan, N.I.; et al. A comprehensive review on emerging importance and economical potential of medicinal and aromatic plants (MAPs) in current scenario. *Pak. J. Agri Res.* **2021**, *34*, 381–392. [CrossRef]
2. Ganguly, S.; Praveen, K.P.; Para, P.A.; Sharma, V. Medicinal properties of chili pepper in human diet. *AJPHCM* **2017**, *2*, 6–7.
3. Shi, Z. Chilli intake is inversely associated with hypertension among adults. *Clin. Nutr. ESPEN* **2018**, *23*, 67–72. [CrossRef]
4. Food and Agriculture Organization of the United Nations (FAO). Production (t). FAO Statistics Database. Available online: https://www.fao.org/statistics/en (accessed on 18 December 2023).
5. da Costa Ribeiro, C.S.; Amaro, G.B.; Reifschneider, F.J.B.; Carvalho, S.I.C. Cultivars. Embrapa Hortaliças. Available online: https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/pimenta/pre-producao/caracteristicas/cultivares (accessed on 18 January 2024).

6. Lillywhite, J.; Robinson, C. Understanding chile pepper consumers' preferences: A discrete choice experiment. *Agriculture* **2023**, *13*, 1792. [CrossRef]

7. Johnson, N.A.N.; Adade, S.Y.S.S.; Ekumah, J.N.; Kwadzokpui, B.A.; Xu, J.; Xu, Y.; Chen, Q. A comprehensive review of analytical techniques for spice quality and safety assessment in the modern food industry. *Crit. Rev. Food Sci. Nutr.* **2025**, 1–26. [CrossRef]

8. Huynh, Q.K.; Nguyen, C.N.; Vo-Nguyen, H.P.; Tran-Nguyen, P.L.; Le, P.H.; Le, D.K.L.; Nguyen, V.C. Crack identification on the fresh chilli (Capsicum) fruit destemmed system. *J. Sens.* **2021**, *1*, 8838247. [CrossRef]

9. Zhong, Y.; Cheng, Y.; Ruan, M.; Ye, Q.; Wang, R.; Yao, Z.; Zhou, G.; Liu, J.; Yu, J.; Wan, H. High-throughput SSR marker development and the analysis of genetic diversity in Capsicum frutescens. *Horticulturae* **2021**, *7*, 187. [CrossRef]

10. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn.* **2021**, *33*, 6999–7019. [CrossRef]

11. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl.* **2023**, *5*, 1680–1716. [CrossRef]

12. Yue, X.; Li, H.; Song, Q.; Zeng, F.; Zheng, J.; Ding, Z.; Kang, G.; Cai, Y.; Lin, Y.; Xu, X.; et al. YOLOv7-GCA: A Lightweight and High-Performance Model for Pepper Disease Detection. *Agronomy* **2024**, *14*, 618. [CrossRef]

13. Dai, M.; Sun, W.; Wang, L.; Dorjoy, M.M.H.; Zhang, S.; Miao, H.; Han, L.; Zhang, X.; Wang, M. Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks. *Front. Plant Sci.* **2023**, *14*, 1230886. [CrossRef]

14. Aguiar, F.P.L.; Nääs, I.A.; Okano, M.T. Bridging the Gap Between Computational Efficiency and Segmentation Fidelity in Object-Based Image Analysis. *Animals* **2024**, *14*, 3626. [CrossRef] [PubMed]

15. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved YOLOv4-tiny model. *Comput. Electron. Agr.* **2021**, *191*, 106503. [CrossRef]

16. Paul, A.; Machavaram, R.A.; Kumar, D.; Nagar, H. Smart solutions for capsicum Harvesting: Unleashing the power of YOLO for Detection, Segmentation, growth stage Classification, Counting, and real-time mobile identification. *Comput. Electron. Agr.* **2024**, *219*, 108832. [CrossRef]

17. Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process Agric.* **2020**, *7*, 1–19. [CrossRef]

18. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]

19. Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hemanth, J. Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy* **2022**, *12*, 2395. [CrossRef]

20. Li, X.; Wang, L.; Miao, H.; Zhang, S. Aphid Recognition and Counting Based on an Improved YOLOv5 Algorithm in a Climate Chamber Environment. *Insects* **2023**, *14*, 839. [CrossRef]

21. López-Barrios, J.D.; Escobedo Cabello, J.A.; Gómez-Espinosa, A.; Montoya-Cavero, L.-E. Green sweet pepper fruit and peduncle detection using Mask R-CNN in greenhouses. *Appl. Sci.* **2023**, *13*, 6296. [CrossRef]

22. Ioannou, K.; Myronidis, D. Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks. *Sustainability* **2021**, *13*, 5323. [CrossRef]

23. Arnob, A.S.; Kausik, A.K.; Islam, Z.; Khan, R.; Rashid, A.B. Comparative result analysis of cauliflower disease classification based on deep learning approach VGG16, inception v3, ResNet, and a custom CNN model. *Hybrid. Adv.* **2025**, *10*, 100440. [CrossRef]

24. Roy, P.S.; Kukreja, K.V. Vision transformers for rice leaf disease detection and severity estimation: A precision agriculture approach. *J. Saudi Soc. Agric. Sci.* **2025**, *24*, 3. [CrossRef]

25. Hussain, M. YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO. *IEEE Access* **2024**, *12*, 42816–42833. [CrossRef]

26. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural Object Detection with You Only Look Once (YOLO) Algorithm: A Bibliometric and Systematic Literature Review. *Comput. Electron. Agr.* **2024**, *223*, 109090. [CrossRef]

27. Ren, R.; Zhang, S.; Sun, H.; Gao, T. Research on pepper external quality detection based on transfer learning integrated with convolutional neural network. *Sensors* **2021**, *21*, 5305. [CrossRef]

28. Kurtulmuş, F.; Alibaş, İ.; Kavdır, I. Classification of pepper seeds using machine vision based on neural network. *Int. J. Agric. Biol. Eng.* **2016**, *9*, 51–62. [CrossRef]

29. Zeng, Y.; Zhao, Y.; Yu, Y.; Tang, Y.; Tang, Y. Pepper disease detection model based on convolutional neural network and transfer learning. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *792*, 012001. [CrossRef]

30. Mohi-Alden, K.; Omid, M.; Soltani Firouz, M.; Nasiri, A. Design and evaluation of an intelligent sorting system for bell pepper using deep convolutional neural networks. *J. Food Sci.* **2022**, *87*, 289–301. [CrossRef]

31. Thakur, P.; Chug, A.; Singh, A.P. Plant Disease Detection of Bell Pepper Using Transfer Learning over Different Models. In Proceedings of the 8th International Conference on Signal Processing and Integrated Networks (SPIN 2021), Noida, India, 26–27 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 384–389. [CrossRef]

32. Antunes, S.N.; Okano, M.T.; Nääs, I.A.; Lopes, W.A.C.; Aguiar, F.P.L.; Vendrametto, O.; Fernandes, J.C.L.; Fernandes, M.E. Model Development for Identifying Aromatic Herbs Using Object Detection Algorithm. *AgriEngineering* **2024**, *6*, 1924–1936. [CrossRef]

33. Roboflow. Available online: https://roboflow.com/ (accessed on 12 January 2024).

34. Matuck, G.R.; Castro, A.J.A.; da Silva, L.E.; Carvalho, E.G. Facial Recognition With Artificial Intelligence Using The Roboflow Platform. *Rev. Prociências* **2023**, *6*, 114–131. [CrossRef]

35. Frizzi, S.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Sayadi, M. Convolutional neural network for smoke and fire semantic segmentation. *IET Image Process* **2021**, *15*, 634–647. [CrossRef]

36. Yilmaz, B.; Kutbay, U. YOLOv8 based drone detection: Performance analysis and optimization. *Computers* **2024**, *13*, 234. [CrossRef]

37. Ultralytics. Ultralytics YOLOv8m Docs. Available online: https://docs.ultralytics.com/ (accessed on 12 November 2024).

38. Google Colab Pro. Available online: https://colab.google.com (accessed on 12 November 2024).

39. Python, Version 3.10.12. Python Software Foundation. 2023. Available online: https://www.python.org/ (accessed on 12 November 2024).

40. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 8024–8035. [CrossRef]

41. Open Source Computer Vision Library. OpenCV. Available online: https://opencv.org/ (accessed on 15 November 2024).

42. Abdulkareem, I.M.; AL-Shammri, F.K.; Khalid, N.A.A.; Omran, N.A. Proposed approach for object detection and recognition by deep learning models using data augmentation. *Int. J. Online Biomed. Eng.* **2024**, *20*, 5. [CrossRef]

43. Zhang, F.; Fu, L.S. Application of computer vision technology in agricultural field. *Appl. Mech. Mater.* **2014**, *462*, 72–76. [CrossRef]