Master Thesis

Insect Pest Detection and Identification Using YOLOv8 on Tomato Crops

Spiros Millas

University of Cyprus
Department of Computer Science

Department of Computer Science

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

Insect Pest Detection and Identification Using YOLOv8 on Tomato Crops

Spiros Millas

Advisor

Dr Costas Pattichis

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2024

ABSTRACT

Agriculture is pivotal to Cyprus's economy, contributing significantly to the GDP and providing essential benefits in rural development, food security and environmental sustainability. However, insect pests pose a significant threat to productivity. This research investigates the efficacy of the YOLOv8 model in detecting and classifying insect pests on tomato plants, aiming to enhance the field of precision agriculture. The experiments evaluated YOLOv8n,YOLOv8s and YOLOv8m, adressing their performance based on precision, recall, mAP50 and mAP50-95 metrics. Results indicated that that the models performed well in detecting larger insects, and struggled when faced with smaller, less distinct insects. This study highlights the need to adress data availability, which hinders work done in this field of research.

Precision Agriculture, YOLOv8, Pest Detection, Computer Vision, Agricultural Sustainability

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivations

Agriculture plays a vital role in the economy and the health of the population of Cyprus. Statistics currently show 3% of workers are employed in the agriculture sector and in 2023 the sector produced 374.4 million Euros, accounting for 1.65% of the total GDP of Cyprus [5]. According to the Cyprus Department of Agriculture, the agricultural sector contributes 2.4% of the national GDP, with crop production contributing 35% of the total value added [6]. On the other hand, greenhouses only occupy 0.5% of the total cultivated area, mostly because they are considered the most intensive and energy-consuming horticultural systems [6].

It cannot be overstated that beyond the fiscal contributions, agriculture also benefits in terms of rural development, food security, nutrition, biodiversity, and environmental sustainability. Furthermore, according to the EU Common Agricultural Policy (CAP), the first objective is to ensure a fair income to farmers, and the second is to increase competitiveness [7]. As reported for Cyprus for the years 2005 to 2018, the agricultural income per worker is on average about 61% of the average wage in the whole economy. Additionally, Cyprus has the lowest share of young farmers in the total number of farm managers in the EU in 2016 (1.3%) [8].

A further contribution of the agriculture sector, includes its intersection with services and the industrial sectors. A study [9] has found that agriculture intersects with the above mentioned sectors, through the production of raw materials such as natural rubber for tires, cotton for clothing, plant oils which act as bio-lubricants as well as bio-diesel. All of the above demonstrate the importance and effectiveness of the agricultural sector.

Insect pests pose a significant threat to agricultural efforts due to their capacity to destroy crops if left unimpeded [10]. Detection and identification of such pests plays a pivotal role in agricultural pest forecasting, thus providing a layer of security against catastrophic crop failure [11]. Traditionally, these tasks are carried out by agricultural experts with the experience to differentiate amongst pest species [12]. However, due to the technological advancements of computer vision models, researchers have taken interest in the field of Precision Agriculture, and it is believed that detection and identification of pests can be done using Artificial Intelligence.[13]

Precision Agriculture (PA) is the intersection of technology and agriculture, and is closely linked to sustainability, offering significant environmental benefits through the targeted and efficient use of resources such as insecticides [14]. Studies [14] have shown that the targeted application of insecticides can significantly reduce insecticide resistance among crop pest populations. Additionally, this approach promotes the conservation of natural predators, which are essential for controlling harmful pests. Together, these benefits lead to a reduced overall use of insecticides, resulting in cost savings and a decreased environmental footprint. Building on this foundation, our research aims to explore the capabilities of the YOLOv8 model in accurately detecting and classifying destructive pests commonly found on tomato plants in Cyprus.

1.2 Related works

The initial approach to gathering literature for this review, involved using the topic title as a search parameter in academic databases. Additionally, the reference sections of relevant articles were explored to find further applicable studies. The focus was primarily on journal articles and conference papers that discuss insect detection and identification using YOLO models. To refine the search effort, articles were initially skimmed to identify pertinent keywords, which were then used for further database searches. The keywords encompassed terms related to insect detection methodologies, with include image processing, computer vision and YOLO model versions. To ensure the relevance and specificity of the search, keyword combinations including the terms ”YOLO”, ”Insect detection”,”Pest Management”, ”Computer Vision” were employed. This strategy was pivotal in zeroing in on the essential information needed for this review. To ensure the accuracy of the review in regards to methodologies currently being explored, the papers chosen for review were published from 2019 and onwards with only the most cited papers being explored.

Initial modern research [15][16][17] on pest detection and classification focused on using CNNs for insect feature extraction rather than undertaking the painstaking process of hand-crafting features. Data collection methods for pests at the time mainly required on site endeavours, either manually [16] or by utilizing IoT solutions [18]. Energy efficiency was shown to be a pressing matter in some researchers directions [18][19], as real time applications for precision agriculture research are needed for timely response to potential pest threats. Traditional CNN backbone approaches such as Resnet101, VGG16, ZFnet proved to be sufficient for this task, reaching 96% precision on classes deemed to be easier due to the larger size of the object, and around 11% accuracy for some of the smaller classes [16].

Further research on the topic of insect detection and classification started the utilization of YOLO networks[11][20][21][13][22]. In a pivotal study ”Insect Detection and Identification using YOLO Algorithms on Soybean Crop” [20] explored the efficacy of YOLO models in detecting and identifying insect pests within soybean crops.

From the discussion of their results, it was gathered that YOLO v3 struggled with smaller or less distinct insect features. The model showed lower precision in handling images with complex backgrounds or those containing multiple insect classes. Yolov4 Demonstrated improved

Algorithm	mAP@0.5	Precision	Recall	F1 Score
YOLO V3	83	95	75	83.52
YOLO V4	94	99	95	96
YOLO V5	99.5	99.2	97.6	96

Table 1.1: Performance comparison of YOLO V3, V4, and V5 in insect detection.

performance over YOLO v3, due o the advancements in network architecture that enhance detection accuracy. The modifications in the models architecture increased the Mean Average Precision (mAP) Precision, Recall and F1 score by approximately 11% and 4%,20% and 13% respectively. YOLO v5 achieved the highest mAP of 99.5 and an F1 score of 96%, which indicated exceptional precision and reliability in pest detection. Detecting small object is a known and significant challenge [23], due to the fact that objects become difficult to discern from background information. To solve this issue, some researchers chose to implement a multi-modal approach [24], where a second classifier is used to re-identify objects in an image. This approach is only needed when working with older YOLO architectures, where there is a benefit benefit to resource trade-off

An overarching theme in most research being done in this field is the use of transfer learning as a tool in their methodology. Researchers [25][26][16] [27] utilize transfer learning in their work to give their model a head start during training, as normally CNNs require a considerable amount of good quality training data to be trained from scratch[28].

The next study that was chosen to review was titled ”Insect Localization and Detection using Object Detection Techniques”[29]. This study includes great work done on evaluating all the YOLO model configurations, from v5 to v8, using the IP102 dataset [30], which is an extensive dataset containing 19,000 annotated images of 102 classes.

Model	Box(P)	R	mAP50	mAP50-95
YOLOv8 s	0.303	0.354	0.303	0.191
YOLOv8 m	0.303	0.356	0.301	0.192
YOLOv8 l	0.327	0.347	0.302	0.192
YOLOv8 x	0.342	0.329	0.305	0.195

Table 1.2: Comparison of YOLOv8 models with different configurations

From their comparative analysis results YOLOv8 proved to be the best model and it is noted that while the models demonstrated an increase in performance in all metrics except recall, the mAP across all classes only slightly increased as the models increased in depth. The increment in precision and mAP which was observed from the smallest model ”s” to the deeper ”x” do manage to refine detection accuracy. This however comes at a price of slower processing speeds, greater training times and increased resources used[31].

From this literature review, it was understood that using YOLOv8 over previous YOLO versions would be the best approach, as it would alleviate some of the issues stemming from small objects in images. The utilization of transfer learning as a tool will prove to be pivotal as the available data on insects that target tomato plantations is limited. Furthermore, from information regarding the efficient use of resources during training, its understood that using the smaller versions of YOLO would be more suitable for training on limited hardware.

1.3 Background Information

1.3.0.1 Convolutional Neural Networks

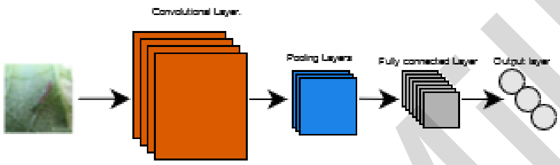


Figure 1.1: typical CNN architecture

Computer Vision is a field of artificial intelligence which deals with image processing and pattern recognition [32]. The primary focus of Computer vision is to create models which extract information from images in the form of patterns of the visible objects present on those images[33]. The most commonly used Neural Network (NN) architecture for computer vision is known as the Convolutional Neural Network (CNN)

The convolution is a linear operation that is used for feature extraction. When an input image is processed by the convolutional layer, a Two-dimensional sliding array of small numbers called a kernel, is applied across the height and width of an image[1]. The image as seen by a computer, is a 2 dimensional array of pixel brightness numbers known as a tensor. With each slide, an element wise product operation is applied between the kernel and input tensor, resulting in a feature map. This operation is visualized on figure 1.2

The resulting feature map is an encoding of the original images distinct features comprised of the underlying patterns observed from the data [34]. This process allows the network to learn kernels which activate when they detect specific features and activations that define the networks ability to recognize patterns in images[35].

Further kernel operations are then done so as to reduce the dimensionality of the convolutional layer output, which helps control overfitting and also reduce the computational cost of the network[36]. After the pooling layer operations, the output is then fed to a fully connected layer which process the feature map to produce an output that is then fed to the output layer which produces a prediction [37]. Figure 1.1 shows a diagram of the simplified CNN network pipeline.

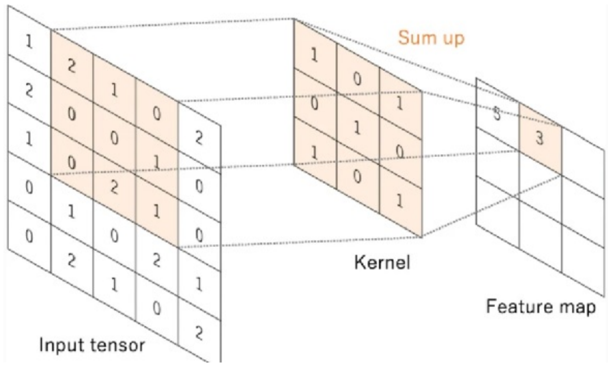


Figure 1.2: Element Wise multiplication [1]

1.3.0.2 Activation functions

Non-linear activation functions play a key role in the training of neural networks such as CNNs. They allow a model to learn complex non linear relationships between inputs and outputs, while also not hampering the training process by controlling the rate of change in the gradient during back-propagation[38].

Several popular non-linear activation functions exist in the literature, such as tanh and logistic sigmoid functions.

$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (1.1) \qquad \mathrm{LogisticSigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (1.2)$$

These activation functions while popular, are computationally expensive to compute thus not very suitable for very deep networks[38]. Furthermore, these activation functions suffer from the problem of vanishing gradient where during training, the gradient becomes too small thus weights are no longer updated, essentially stopping the training procedure [38].

A common non linear activation function being used for deep learning is the Rectified Linear Unit function, abbreviated to ReLU.

$$\mathrm{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1.3}$$

ReLU outputs zero for all negative inputs and acts as the identity for positive inputs. This characteristic results in sparse activations and helps mitigate the vanishing gradient problem during back propagation due to its non-contractive property[39]. Being inherently non-negative, ReLU has a mean activation greater than zero, which leads to a bias shift in future activations that is exacerbated the more the units are correlated [39]. Furthermore, the constant 0 gradients produced by the ReLU function can lead to slower training[40].

In modern CNNs tailored to object detection, the SiLU (also known as SWISH) activation function is used and has the following equation [41]

$$\text{SiLU}(x) = x\left(\frac{1}{1+e^{-x}}\right) \tag{1.4}$$

From the research done by Ramachandran et.al [42], it is understood that it has SiLU has many benefits when training deep networks. Due to the smooth nature of SiLU is provides a stable training procedure when compared to other non linear functions such as ReLU. Unlike ReLU, SiLU maintains a gradient for negative inputs which helps avoid the vanishing gradient problem. SiLU performance has been proven across a range of models and datasets, which demonstrates its robustness. These characteristics make this activation function suitable for training large scale models of varying depths, which may be the reason researchers at ultralytics chose it as part of their model architecture for YOLOv8[2].

1.3.0.3 RCNN history, YOLO characteristics and YOLOv8 architecture

In the realm of computer vision, the objective of object detection extends beyond merely recognizing whether objects belong to general categories within an image. It also involves precisely pinpointing their spatial locations and defining their boundaries using bounding boxes[43]. Modern object detection began with the proposal of R-CNN which utilized a three stage detection process where an image is first split into proposed regions, those regions being processed individually using a CNN, then classification is done using class specific SVMs [44]. This processed was built upon with the proposal of the Fast-RCNN where the detection process is reduce to a single stage. Taking as input an image and a set of object proposals, the entire image is processed by a CNN and the region of interest is processed by an RoI pooling layer that feeds into several fully connected layers which lead to a sibling out output layers consisting of a softmax layer and a bounding box regressor. [45]. In the last improvement to the original RCNN, the detection process is sped up by utilizing a Region Proposal Network in the main architecture. The RPN takes as input the feature map generated by the CNN and produces regions of interest which are pooled and classified using the Fast-RCNN detector[46]. This last network is known as the Faster -RCNN.

YOLO transforms the object detection procedure by framing it as a single regression by framing it as a single regression problem. The model predicts bounding boxes and class probabilities directly from full images using CNNs. During training and inference, an input image is divided into M number of grids with each grid having equal SxS dimensional regions. Each grid is populated by a number of bounding boxes with coordinates relative to cell coordinates [47]. Each bounding box consists of the (x,y) box center coordinates relative to the grid cell, the height and width of the box relative to the input image and the confidence score associated with the prediction which represents the IoU score between the box and he ground truth label[48].

Over the years, the YOLO network has went through many changes between versions. Currently, the latest version published by Ultralytics is the 8th version of YOLO, apptly named YOLOv8 [4].

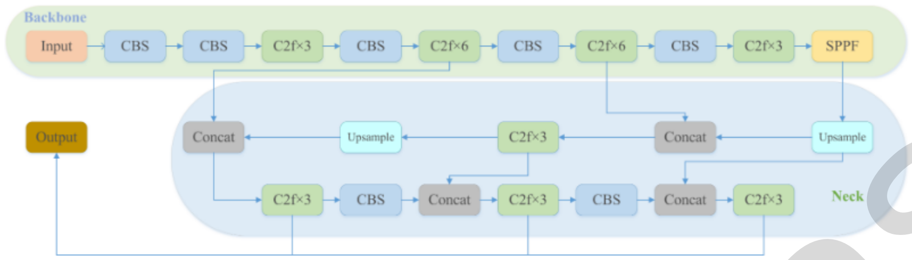


Figure 1.3: Architecture of the YOLOv8 Network [2]

The Yolov8 backbone utilizes the CSPDarknet53 structure, from which it takes inspiration in the design of the C2F modules [2][49]. The CSPnet is used due to its ability to enhance CNN learning capabilities by partitioning feature maps into two pars and merging them through a cross-stage hierarchy. This design has been proven to improve computational efficiency and enhancing accuracy[50]. The convolution kernels placed in front of the C2F module are known as the CBS modules, and consist of a 3x3 convolutional layer with stride = 2, batch normalization which improves the training of neural networks y stabilizing the distributions of layer inputs[51], and the SiLU activation function. These layers serve the purpose of down sampling the input. The output of the CBS modules is then fed to a number of C2F modules which consist of two convolutional layers and cross-stage partial bottleneck (Figure 1.4).

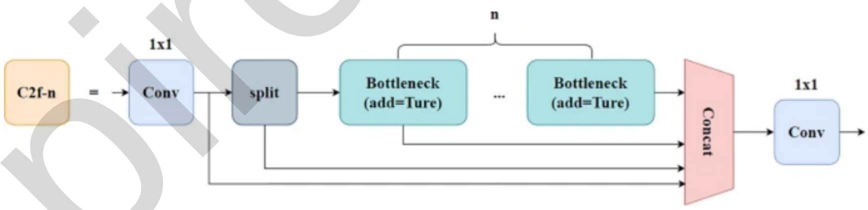


Figure 1.4: C2F module architecture [3]

At the last layer of the backbone, a spatial pyramid pooling fast layer (SPPF) is utilizes. The SPP layer enhances CNNs by allowing them to accept any input size without requireing image resizing, which preserves spatial information. SPP generates fixed length representations regardless of image size which makes the network preform well even when target objects are deformed and vary in scale[52]. The SPPF architecture can be seen on Figure 1.5

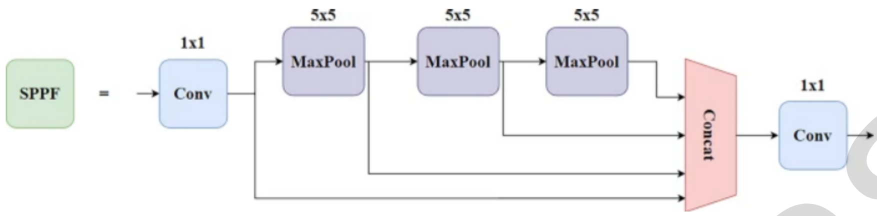


Figure 1.5: SPPF module architecture [3]

Acting as a bridge between the Backbone and head of the network, the YOLOv8 neck preforms feature fusion operations and integrates the contextual information by assembling feature pyramids and aggregating the maps obtained at different stages of backbone operations. The neck essentially fuses features of different scales to help the model detect objects at different sizes[3]

The last portion of the YOLOv8 network is the head, which is responsible for generating the output of the network consisting of Bounding boxes, and confidence scores associated with those bounding boxes

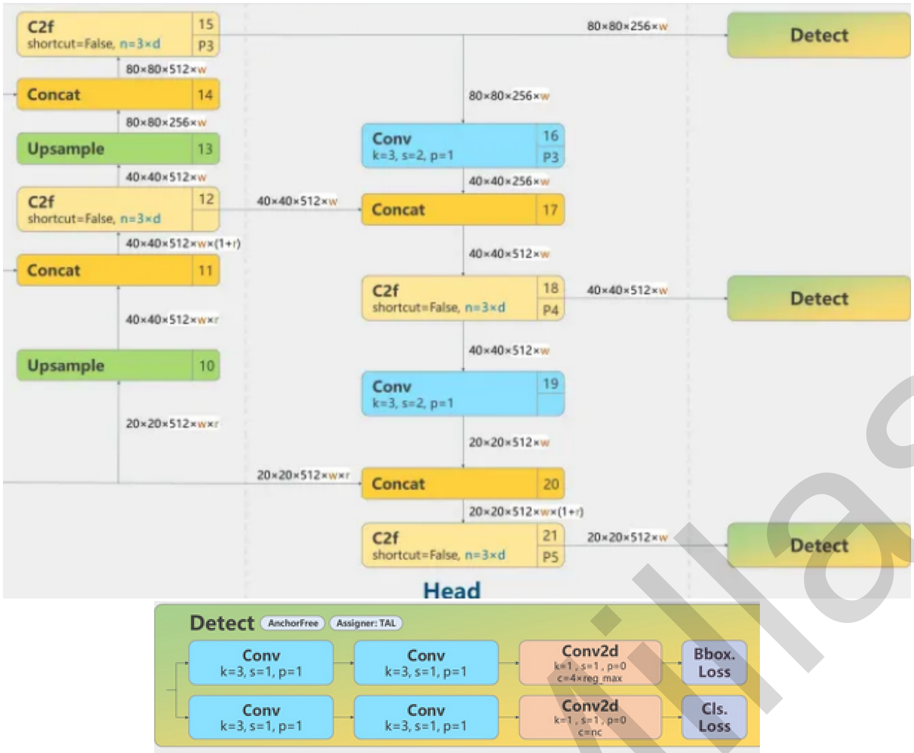


Figure 1.6: YOLOv8 Head architecture [4]

This section consists of three detection heads with each head corresponding to a separate task. These tasks include classification, bounding box prediction and regression. For classification, the Binary cross entropy loss function is used, which is a popular multi-class classification function, while for the bounding box regressor, the CIoU loss and DFL funcitons are utilized[53].

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}[54], \tag{1.5}$$

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v.[54] \tag{1.6}$$

$$DFL(S_i, S_{i+1}) = -\left((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})\right)[55] \tag{1.7}$$

$$\mathcal{BCE} = -\left(y_i \cdot \log(p(x_i)) + (1 - y_i) \cdot \log(1 - p(x_i))\right)[55] \tag{1.8}$$

1.4 Summary

Agriculture is crucial to Cypru's economy, contibuting significantly to GDP and providing essential benefits such as food security, development to rural areas and environmental sustainability. However, insect pests pose a significant threat to agricultural productivity. Traditionally, pest detection and identification relies on expert knowledge, but advancements in computer vision has allowed artificial intelligence to be integrated to the research on PA. Utilizaiton of these technologies on a wider scale can help in improving resource efficiency, reduce insecticide resistance and conserve natural insect predators. This research leverages the use of the state of the art YOLOv8 model to enhance pest detection and classification in tomato plantations.

Section 1.2 highlights the evolution of pest detection methodologies, from early standard CNN based approaches for classification, to the adoption of object detection models such as YOLO. Modern studies were shown to emphasize the performance of YOLO over other object detection models, with some already assessing the current SOTA models in tackling issues such as complex backgrounds and small insect detection. Also from the literature review, transfer learning was identifies to be a crucial tool for training such models efficiently, leveraging past knowledge for fine tuning on insect datasets which are sparse due to lack on information.

2 Methodology

2.1 Materials and Methods

2.1.1 Data Collection and Processing

2.1.1.1 Data Collection

As stated in section 1.0, the focus of this experiment was on insect pest which affect tomato plants in Cyprus. Spefically, Aphid, Leafhoppers, Spider Mites, Spodoptera Larva, Spodoptera moth, Stinkbug, Thrips. The pests were chosen after taking into consideration the advice given by a prominent professor from the Cyprus Technological University, and also the availability of public data on the internet. The data was primarily sourced from online databases of insect pests, forums where users post images and details of identified pests. This comprehensive approach ensured the data that was gathered was relevant and the insects that were included in the dataset were correctly identified.

Figure 2.1: 1: Spodoptera Larva, 2: Red Spider Mite, 3: Thrips, 5: Spodoptera Moth, 6: Southern Green Stinkbug, 7: Aphids

Figure 2.1 shows a mosaic of images, showcasing the various targeted insects. Each of the targeted insect pests presented has a unique way to hinder plant life by either eating foliage, draining nutrients from the host plant through the feeding on its sap and spreading plant diseases [56].

2.1.1.2 Data Preprocessing

The data pre-processing for this experiment was conducted using the robust online platform Roboflow. Roboflow was designed for managing and preparing datasets for machine learning, by providing an intuitive interface for labeling, preprocessing steps and data augmentation. The dataset was was first split to train validation and test sets, with a split of 85%, 10% and 5%

respectively, making sure to include enough instances of each targeted insect in the final set. Next, the data was resized to 640x640, which is the preferred size for training YOLOv8[4], since the models were pretrained on the COCO dataset which includes images of size 640x480. Furthermore, the data was auto-oriented to ensure that all images are fed to the model in the correct orientation.

Last and most important, several augmentations were done on the training set, which were essential in enhancing the diversity and robustness of the training set. Augmentations to the training set have been shown to increase the overall accuracy of models by allowing the model to learn the underlying features more effectively, and also generalize better to unseen data [57]. The first augmentation that was chosen was a 50% chance to flip an image vertically or horizontally. Flipping augmentations help the model learn that objects can appear in any orientation. For instance, an insect might be facing different directions in real world scenarios, and the flipped images introduce engineered samples of those situations to the model. Likewise, rotations of the images serve the same purpose by introducing variability in the angles which the objects are presented to the model. The rotations were randomly applied at 90 degrees clockwise, 90 degrees counter-clockwise, and 180 degrees. Furthermore, slight tilts were applied to the images equally randomly, at between +- 15%, to simulate the slight tilts and that can occur when photos are taken from handheld devices. The last augmentation step was the addition of random blur effect to the images up to a 2.4px radius, which simulates pictures taken without a properly focused camera. This is advantageous as some of the insect pests are quite small, meaning that a camera would have to be positioned quite close to the subject which hinders the focusing ability.

2.1.1.3 Model Selection and Training

As stated above, the chosen model for this experiment was Yolov8, which as of January 2024, is the state of the art YOLO model architecture. Given that the dataset that was used to train the model was relatively small, only containing 3063 images in the training set across seven classes, transfer learning was utilized to leverage pre-existing knowledge. To preform a comprehensive evaluation of Yolov8's capabilities for this task, multiple petrained versions of the network were put to the task. Specifically, the Yolov8 nano, small and medium models were tested. Training was conducted on an NVIDIA 1070 GPU, which while capable, has limited memory and computational power compared to more modern GPUs. One thing to note is that Yolov8 has five pre-trained model architectures available, with the last two being the large and extra large versions. These two model versions are very resource intensive and as a consequence of the hardware being used, could not be utilized for this experiment. The training process involved fine-tuning the pre-trained models on the custom dataset using WanDB to monitor the training procedure and generate graphs related to the performance of the model.

Optimizer	Momentum	Learning Rate	Learning Rate schedule
AdamW	0.9	0.1	0.01

Table 2.1: Hyperparameters used in the experiment

Yolov8 has several hyperparameters. the most major the most major one being the optimizer, which is the mechanism that adjusts the model's parameters during training, so as to minimize the loss function and ultimately improve accuracy. For this experiment, the AdamW optimizer was chosen. AdamW works by decoupling weight decay from the gradient update process, which is shown to have better results than the preceding Adam optimizer, leveraging the benefits of faster convergence of Adam but also allowing for greater generalization abilities [58]. The other important hyperparameters are listed in Table 3. These were used because they are the default settings recommended by Ultralytics.

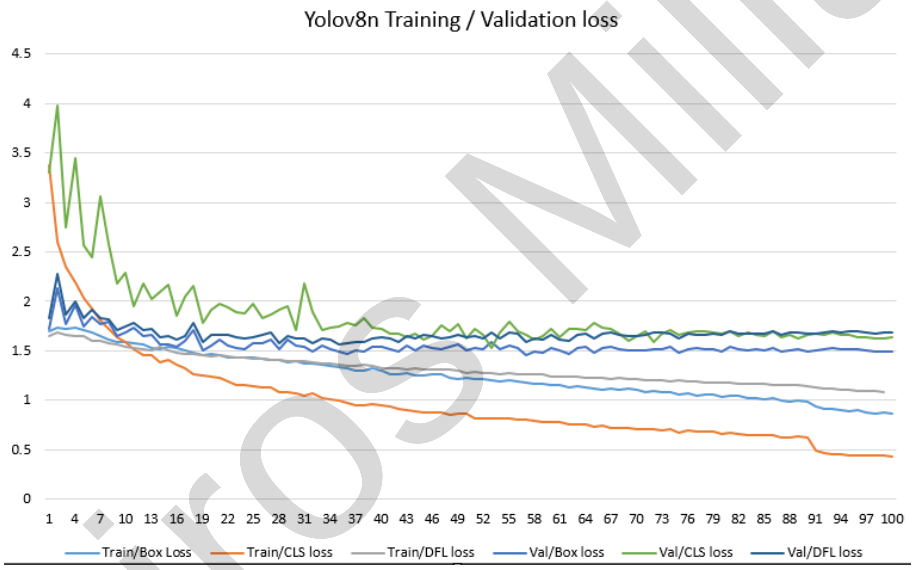


Figure 2.2: Yolov8n model training losses

Figure 2.2 illustrates the training and validation loss plots gathered during the fine tuning process of the Yolov8n pretrained model over 100 epochs, including Classifier, DFL and Bounding box losses. The training time for each model was equally for 100 epochs, with the classifier loss being the most important metric to determine the optimal training time. As shown on the figure, the train classifier loss begins to plateau and for the validation classifier loss, the losses begin to increase, indicating potential overfitting. After the training procedure finishes, the best weights are selected to represent the fine tuned model.

2.1.2 Evaluation Metrics

2.1.2.1 Precision

Precision is the ratio of true positive detections to the sum of true positives (TP) and false positives (FP). It measures the accuracy of the positive predictions made by the model.

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

2.1.2.2 Recall

Recall is the ratio of TP detections to the sum of TP and FP. It measures the model's ability to correctly identify all objects in the dataset.

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

2.1.2.3 mAP50

mAP50 is the mean Average Precision calculated at an Intersection over Union (IoU) threshold of 0.5. Detections are considered as TP if the IoU between the predicted and ground truth labeled bounding boxes is at least 0.5.

The Average Precision (AP) is calculated for each class as the area under the precision-recall curve. mAP50 is the mean of the AP values across all classes (N).

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

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall})$$

$$\text{mAP50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

2.1.2.4 mAP50-95

mAP50-95 is the mAP calculated at multiple IoU thresholds, from 0.5 to 0.95 with a step size of 0.05. This metric provides a more comprehensive evaluation of the model's performance across different IoU thresholds

$$\text{mAP50-95} = \frac{1}{10N}\sum_{i=1}^{N}\sum_{t=0.5}^{0.95}\text{AP}_i(t)$$

3 Results

3.1 Experiment Results

In this section, the outcomes of the experiments will be presented, with a focus on evaluating the YOLOv8 model's performance in detecting and classifying insect pests on tomato plants. A detailed analysis of the models training and validation performance, showcasing the effectiveness of the tested versions. The results will be evaluated based on the metrics mentioned in section 2.2, which will also include comparisons between the versions with a focus on benefits and drawbacks. Further comparisons will be done with works from other researchers on older YOLO model versions, so as to assess whether there was an increase in performance with the use of YOLOV8.

3.1.1 YOLOv8n experiment results

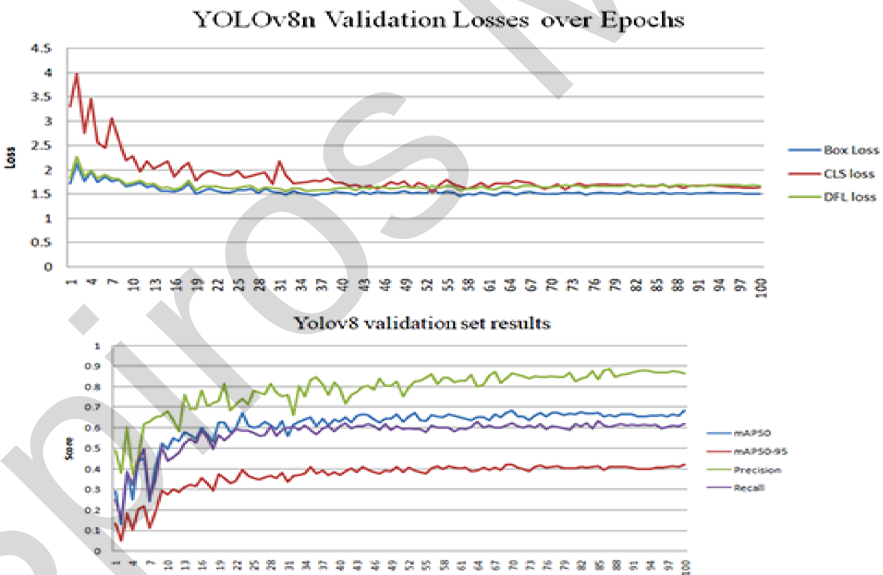


Figure 3.1: Yolov8n model training losses

Figure 3.1 illustrates the validation set losses for 3 different metrics, those being the Box, Classifier and DFL losses across the 100 epoch training process. Initially all 3 metrics show a sharp decline then plataeu near the end of the training process, indicating that the model is in fact learning to classify object and to predict bounding box location. Also shown on the figure , are the performance metrics on the validation set associated with the training procedure. As the model progresses through its training, the metrics are shown to increase, eventually plateauing.

Class	P	R	mAP50	mAP50-95
all	0.679	0.496	0.585	0.348
Aphid	0.856	0.0832	0.284	0.134
Leafhopper	0.723	0.762	0.956	0.553
Spider Mite	0.166	0.250	0.0884	0.0838
Spodoptera Larva	0.652	0.375	0.481	0.301
Spodoptera Moth	0.921	0.776	0.928	0.605
Stinkbug	0.610	0.667	0.702	0.374
Thrips	0.689	0.640	0.658	0.387

Table 3.1: Performance Metrics for YOLOv8n Model on the test set

Table 4 presents the results gathered after evaluating the performance of the model on the test set, which highlight its strengths and limitations. The overall metrics indicate a decent precision percentage of 67.9%, with an average recall of 49.6%. With some classes such as the Spodoptera moth and leafhoppers being exceptionally good, which is to be expected due to their large size and very distinct shape compared to background information. This fact is further confirmed with the large overlap accuracy of the predicted box and the ground truth label, with a map50 of 92.8%/95.6% and an map50-95 of 60.5%/55.3% respectively. However, when dealing with classes that are much smaller in size compared to the background information, the model was shown to lack the ability to spot the insect present in those images.

This fact is confirmed when looking at the aphid class specifically, where there is a high precision when the model actually manages to detect the object, but a very low recall rate, indicating that most of the instances of aphids are left undetected. Furthermore, as aphids are usually present in clusters, the map50 and map50-95 scores are quite low, indicating that the model has trouble placing the bounding box around the insects. Furthermore, when looking at spiter mites, which are an equally small insect when compared to aphids, the model was shown to preform very poorly with a very small precision and an equally small Recall percentage.. The rest of the classes fall somewhere in the middle in terms of performance, with some classes like thrips being correctly identified around 51% of the time, but being detected at reasonable rates. Lastly, like thrips, spodoptera larva fall somewhere in the middle performance wise, with a precision of 60.6% and a recall score of 51.2%. Comparing thrips and larva on the metric of mAP, an interesting distinction can be made, where because of the color, shape and orientation of the larva, the model may have a harder time predicting a bounding box.

Class	P	R	mAP50	mAP50-95
all	0.865	0.618	0.682	0.422
Aphid	1.000	0.0125	0.135	0.0687
Leafhopper	0.936	0.734	0.869	0.519
Spider Mite	0.832	0.316	0.395	0.189
Spodoptera Larva	0.909	0.762	0.860	0.545
Spodoptera Moth	0.952	1.000	0.984	0.686
Stinkbug	0.742	0.755	0.775	0.489
Thrips	0.682	0.746	0.754	0.455

Table 3.2: Performance Metrics for YOLOv8n Model on the validation set

Table 5 presents the results gathered after evaluating the performance of the YOLOv8n model on the Validation set at the final epoch. When compared to table 4, its clear that either an element of overfitting or a poor generalization ability is affecting this models performance, as indicated by a difference in the overall precision and recall metrics across different classes. What is most evident, is the difference in the spider mite class where there exists a 66.6% difference in precision which may be due to the difference in appearance of the spider mites in terms of angles or the difference in background clutter present in those images between the validation and test sets.

3.1.2 Yolov8s Experiment results

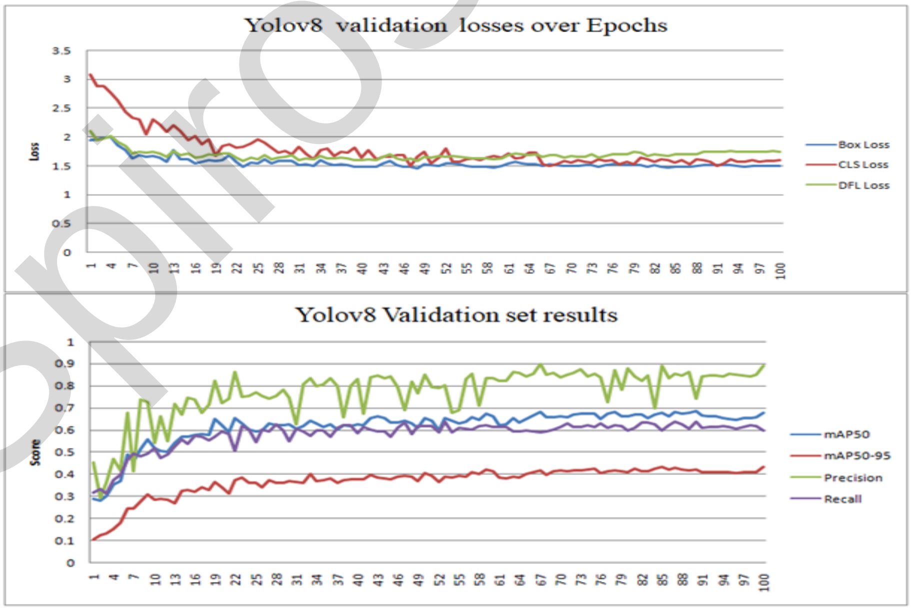


Figure 3.2: Yolov8s model training losses

Figure 3.2 illustrates the validation set losses for three different training evaluation metrics Box, Classifier, and DFL losses, across the 100-epoch training process of the YOLOv8s model. As expected, all three metrics showed a decline, indicating that the was learning. The decline was followed by a plateau towards the end of the training process, which suggested that the model had effectively stabilized. The figure also displays the performance metrics on the validation set, which show an incline that suggested the model was improving during training, eventually plateauing towards the end.

Class	**Precision**	**Recall**	**mAP50**	**mAP50-95**
All	0.695	0.584	0.603	0.382
Aphid	1	0.00782	0.364	0.18
Leafhopper	0.827	0.824	0.887	0.509
Spider Mite	0.0925	0.25	0.0428	0.0385
Spodoptera Larva	0.681	0.458	0.458	0.329
Spodoptera moth	1	0.827	0.933	0.662
Stinkbug	0.634	1	0.849	0.529
Thrips	0.631	0.721	0.688	0.423

Table 3.3: Performance Metrics for YOLOv8s Model on the test set

Table 3.3 shows the performance metrics gathered after YOLOv8s was evaluated on the test set. Firstly, an increase in all metrics across all classes was evident, with a roughly 2%, 9%, 2% and 3% increase in precision, recall, mAP50 and mAP50-95 respectively. Digging deeper in the results, by looking at each class individually. some where shown to exhibit either unrealistic Precision and recall scores, specifically Spodoptera moths and stinkbugs. This may be due to the fact that the test set was too small to properly validate the performance of this model for those specific metrics. However, mAP was also shown to increase across the board for almost all classes, indicating that the more complex model managed to fit bounding boxed more accurately.

Class	**Precision**	**Recall**	**mAP50**	**mAP50-95**
All	0.889	0.6	0.68	0.435
Aphid	1	0	0.113	0.0629
Leafhopper	0.938	0.761	0.895	0.57
Spider Mite	0.738	0.342	0.388	0.196
Spodoptera Larva	1	0.653	0.879	0.562
Spodoptera moth	0.957	1	0.993	0.704
Stinkbug	0.734	0.735	0.71	0.456
Thrips	0.856	0.71	0.784	0.495

Table 3.4: Performance Metrics for YOLOv8s Model on the validation set

When comparing the test set results to the validation set, issues regarding generalization are again observed, indicating that the more complex model did not manage to improve in that aspect. This discrepancy suggests that the model may be overfitting, capturing noise and specific patterns that do not generalize well to unseen data. Consequently, the model's ability to accurately predict and classify objects in a real-world scenario is compromised, reflecting a gap between training/validation performance and practical application.

3.1.3 YOLOv8m experiment results

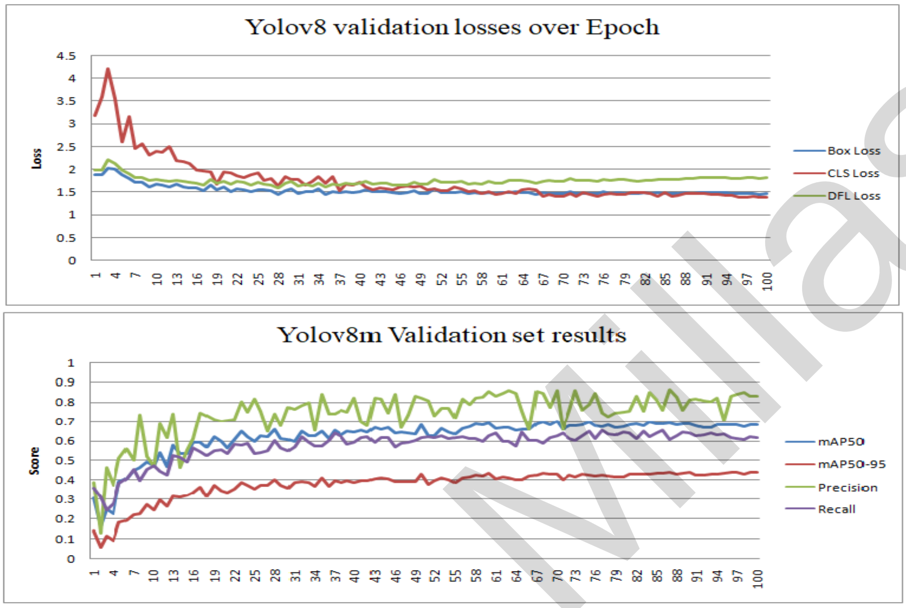


Figure 3.3: Yolov8m model training losses

Class	Precision	Recall	mAP50	mAP50-95
all	0.673	0.546	0.61	0.396
Aphid	0.919	0.0509	0.346	0.168
Leafhopper	0.816	0.882	0.956	0.548
Spider Mite	0	0	0.00419	0.00377
Spodoptera Larva	0.714	0.417	0.472	0.343
Spodoptera moth	0.975	0.833	0.903	0.661
Stinkbug	0.72	1	0.939	0.648
Thrips	0.568	0.642	0.646	0.403

Table 3.5: Performance metrics for different classes

Table 3.5 presents the results gathered after evaluating the performance of the YOLOv8m model on the test set. Being the largest model that was trained in this experiment, the results were expected to be the best out of all the models. However, when compared to the results gathered from the previously trained smaller models, YOLOv8m did not show improvement. On the contrary, every metric across the board seemed to have lowered. This may indicate that the model has overfit on the limited data. To support this finding, a comparison with the validation data was done with findings similar to previous models.

Class	Precision	Recall	mAP50	mAP50-95
all	0.833	0.616	0.684	0.44
Aphid	1	0	0.101	0.045
Leafhopper	0.953	0.778	0.904	0.584
Spider Mite	0.572	0.342	0.402	0.169
Spodoptera Larva	0.94	0.69	0.868	0.592
Spodoptera moth	0.955	1	0.989	0.695
Stinkbug	0.734	0.816	0.792	0.51
Thrips	0.676	0.681	0.736	0.487

Table 3.6: Performance Metrics for YOLOv8m Model on the validation set

3.2 Conclusion

In this section, the results of the experiments using various YOLOv8 models (YOLOv8n, YOLOv8s, and YOLOv8m) for detecting and classifying insect pests on tomato plants were presented and analyzed. The focus was on evaluating each model's performance based on key metrics such as precision, recall, mAP50, and mAP50-95

3.3 YOLOv8n Results

Yolov8n showed promising results, with a balanced performance across different classes. The model preformed well when having to detect and classify larger insects such as stinkbugs, Spodoptera moths and leafhoppers. However, the model preformed poorly when having to detect the smaller classes such as aphids and spider mites. The model exhibited some issues with generalization, but that is to be expected to some extend as the dataset has high variability in the way insects are presented in the images.

3.3.1 YOLOv8s Results

The YOLOv8s model showed an overall improvement in performance metrics compared to the YOLOv8n model. This improvement was reflected in higher precision, recall, mAP50, and mAP50-95 across most classes. However, some unrealistic precision and recall scores for classes like Spodoptera moths and stinkbugs suggest that the test set might not have been sufficiently comprehensive. The increase in mAP scores across the board indicates better bounding box accuracy, although generalization issues persisted.

3.3.2 YOLOv8m Results

Contrary to expectations, the YOLOv8m model, being the most complex and parameter-heavy, did not outperform the smaller YOLOv8 models. In fact, the model's performance metrics decreased across all classes. This suggested that the model might be over-parameterized, leading to overfitting and poorer generalization to unseen data. The validation set results further confirmed this, highlighting the gap between training/validation performance and real-world application.

3.3.3 General Observations

Across all models, smaller insects such as aphids and spider mites consistently posed challenges, likely due to their size and less distinct features compared to the background. Larger insects like Spodoptera moths and Leafhoppers, which have more distinctive shapes and sizes, were detected and classified more accurately. The discrepancies between validation and test set performances across all models point to a need for generalization techniques and more diverse training data to improve real-world applicability. Furthermore, when assessing the deeper models, YOLOv8s did show better results than its smaller counterpart, but at a cost to a 53% increase in training time, which is significant. YOLOv8m hardly preformed better than the shallowest network, indicating that for this experiment and possibly for others in this area of research, there is no need to utilize the deeper and more resource demanding networks, especially when currently the available data is limited.

4 Discussion/ Interpretation

4.1 Performance discussion

When looking at the results gathered from the evaluation of these models, it is evident that YOLOv8 has the capabilities to correctly detect and identify larger insect pests. The models all excel in detecting larger insects, but struggle with smaller less distinct insects. One reason for this phenomena is the amount of training data of those specific insects fed to the model during training. While some insects such as stinkbugs, spodoptera moths, larva and thrips have ample training data available online for free use, other insects such as spider mites have very limited realistic data. The limited training data for smaller insects means the models have fewer examples to learn from, leading to lower precision and recall rates for these classes.

Additionally, the inherent characteristics of smaller insects pose a significant challenge for detection algorithms, due to the fact that they often blend into the background due to their size, making it difficult for the model to distinguish them from the environment surrounding them. This issue is further compounded by the variability in image quality and lighting conditions, which can obscure the already subtle features of these tiny pests. In summary, the current state of the art YOLO model still suffers from the same issues that its predecessors have when having to identify small objects. This, coupled with the fact that computer vision in the field of precision agriculture is still developing in terms of data availability, means that any further research done in this field requires a comprehensive real world data gathering campaign, and a multi-modal approach to the classification process if targeting smaller insects.

4.2 Qualitative analysis

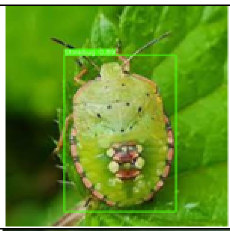
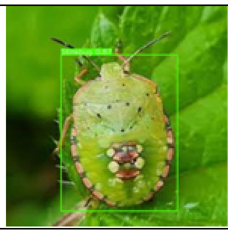
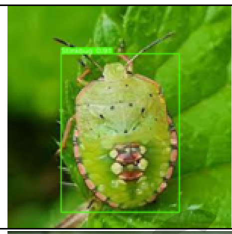
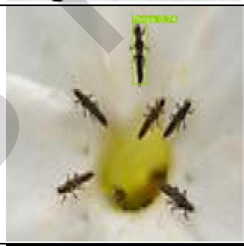
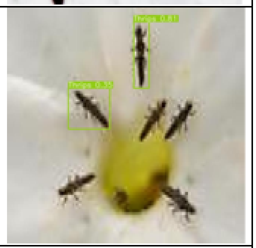
Class	YOLOv8n	YOLOv8s	YOLOv8m
Stinkbug			
Spodoptera Larva			
Spodoptera Moth			
Thrips			
Leafhopper			

Figure 4.1: Qualitative Analysis

Figure 4.1 illustrates the output of the different YOLOv8 model versions on randomly sampled images of insect classes where the model preformed satisfactory. Firstly, all three models demonstrated high accuracy in detecting these insects, with no noticeable differences in their output. This consistency leads to the conclusion that there may be no need to utilize the larger models for such a task, which can save valuable computational resources for any researchers down the line. All models had a problem when having to detect multiple objects in a single image, with YOLOv8n seeming to have the best output, managing to capture the same insects as its larger counterparts, and making an effort to capture the third, but not managing to do so accurately due to the orientation of the insects.

4.3 Comparison with related works

Comparing the outcomes of this experiment with other related works found in the literature, presents its own problems. Research done in this specific area exhibits a high variability with regards to the target insects, which as proven in section 3, plays a great role to the overall performance of the models being utilized. Nonetheless, when comparing YOLOv8 results on tomato plants to those gathered from [20], where researchers used YOLOv5 out of the box, an interesting finding emerges. From their quantitative results, it seems that their proposed approach was outperformed by YOLOv8 in terms of precision, but when looking at the actual recall score, their model captured more total insects.

Further comparisons of the experiments can be done with researchers who chose to use transformer based computer vision architectures for the task of pest detection and and identification. One study[59] used a proposed Enhanced Vision Transformer Architecture for the task of pest image segmentation and classification, where they achieved precision and recall scores of 92% and 96% respectively. Their results highlight the promising potential of newer architectures for object detection, attributed to the transformers architecture's ability to process and learn from image data with high variability and complexity. It is also noted that their experiments included images of difficult to detect insects, specifically the aphid.

From the information gathered from this experiment, it is clear that without modificaitons to the algorithms, YOLOv8 still struggles with detecting small objects. Comparing the results on small insects from the out of the box best performing YOLOv8 model to the research done by [24], it is evident that their two-step classification approach would be beneficial to adopt for researchers down the line. For example, the researchers managed to improve YOLOv3's performance by more than 60% when having to detect insects equally as small and indiscernible as aphids.

5 Summary of Findings/ Recommendations

The evaluation of the YOLOv8 models demonstrated the model's capabilities and limitations in detecting and classifying insect pests on tomato plants. Each model showed proficiency in detecting larger and more distinct insects, which can be attributed to their large amount of available training data, but also their inherent features which are easier for the models to recognize and classify.

However, all models faced significant challenges when detecting smaller insects such as aphids and spider mites. These difficulties are mostly due to the limited amount of training data available for these specific insect classes, which coupled with the less distinct features, made the detection and classification process very poor. Quantitative analysis confirmed these findings, showing consistent detection performance across all models for larger insects, while smaller insects posed a challenge.

These findings highlight the need for more comprehensive real-world data collection and potential adoption of multi-modal approaches to improve the detection of small objects. Future research should focus on enhancing the models' capabilities to generalize better to new data and developing techniques to improve the detection of smaller, less distinct insects. Furthermore, research that includes testing the capabilities of vision transformers for insect pest object detection will be beneficial to the precision agriculture landscape

BIBLIOGRAPHY

[1] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018. [Online]. Available: https://doi.org/10.1007/s13244-018-0639-9

[2] J. Guo, H. Lou, H. Chen, H. Liu, J. Gu, L. Bi, and X. Duan, “A new detection algorithm for alien intrusion on highway,” *Scientific Reports*, vol. 13, 07 2023.

[3] T. Uygun and M. Ozguven, “Real-time detection of shot-hole disease in cherry fruit using deep learning techniques via smartphone,” *Applied Fruit Science*, 2024.

[4] Ultralytics, “Yolov8,” 2023.

[5] A. O’Neill, “Distribution of gross domestic product (gdp) across economic sectors cyprus 2022.”

[6] C. M. of Agriculture. (2022) Rural development and the environment. Accessed 29/9/22. [Online]. Available: https://moa.gov.cy/?lang=en

[7] E. C. Agricultural and R. Development. (2022) Key policy objectives of the new cap. Accessed 29/9/22. [Online]. Available: https://ec.europa.eu/info/food-farming-fisheries/key-policies/common-agricultural-policy/new-cap-2023-27/key-policy-objectives-new-cap_en

[8] European Commission Agricultural and Rural Development. (2022) Cyprus analytical factsheet. Accessed 29/9/22. [Online]. Available: https://ec.europa.eu/info/sites/default/files/food-farming-fisheries/by_country/documents/analytical_factsheet_cy.pdf

[9] M. University, “Why is agriculture important? benefits and its role.”

[10] E.-C. OERKE, “Crop losses to pests,” *The Journal of Agricultural Science*, vol. 144, no. 1, p. 31–43, 2006.

[11] K. Li, J. Zhu, and N. Li, “Lightweight automatic identification and location detection model of farmland pests,” *Wireless Communications and Mobile Computing*, vol. 2021, 10 2021.

[12] L. Deng, Y. Wang, Z. Han, and R. Yu, “Research on insect pest image detection and recognition based on bio-inspired methods,” *Biosystems Engineering*, vol. 169, pp. 139–148, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511017302969

[13] A. Bibi, M. Moetesum, and I. Siddiqi, “Deep vision for pest detection in corn fields,” in *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2022, pp. 343–348.

[14] R. Bongiovanni and J. Lowenberg-DeBoer, “Precision agriculture and sustainability,” *Precision Agriculture*, vol. 5, pp. 359–387, 08 2004.

[15] R. Li, R. Wang, J. Zhang, C. Xie, L. Liu, F. Wang, H. Chen, T. Chen, H. Hu, X. Jia, M. Hu, M. Zhou, D. Li, and W. Liu, “An effective data augmentation strategy for cnn-based pest localization and recognition in the field,” *IEEE Access*, vol. 7, pp. 160 274–160 283, 2019.

[16] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman, and W. Liu, “Pestnet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification,” *IEEE Access*, vol. 7, pp. 45 301–45 312, 2019.

[17] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, “An efficient approach for crops pests recognition and classification based on novel deeppestnet deep learning model,” *IEEE Access*, vol. 10, pp. 73 019–73 032, 2022.

[18] D. Brunelli, A. Albanese, D. d’Acunto, and M. Nardello, “Energy neutral machine learning based iot device for pest detection in precision agriculture,” *IEEE Internet of Things Magazine*, vol. 2, no. 4, pp. 10–13, 2019.

[19] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie, and T. Shongwe, “Internet of things and wireless sensor networks for smart agriculture applications: A survey,” *IEEE Access*, vol. 11, pp. 145 813–145 852, 2023.

[20] S. Verma, S. Tripathi, A. Singh, M. Ojha, and R. R. Saxena, “Insect detection and identification using yolo algorithms on soybean crop,” in *TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)*, 2021, pp. 272–277.

[21] M. T. Farooq, M. Leipert, A. Maier, and V. Christlein, “Ressource-efficient moth detection for pest monitoring with yolov5,” in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023, pp. 553–558.

[22] N. Mamdouh and A. Khattab, “Yolo-based deep learning framework for olive fruit fly detection and counting,” *IEEE Access*, vol. 9, pp. 84 252–84 262, 2021.

[23] W. Wei, Y. Cheng, J. He, and X. Zhu, “A review of small object detection based on deep learning,” *Neural Computing and Applications*, vol. 36, pp. 1–21, 02 2024.

[24] H. Kuzuhara, H. Takimoto, Y. Sato, and A. Kanagawa, “Insect pest detection and identification method based on deep learning for realizing a pest control system,” *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2020.

[25] X. Li, Z. Yue, J. Su, S. Wang, J. Hua, and F. Duan, "Application of lightweight object detection network in cucumber leaf detection," in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2022, pp. 686–691.

[26] M. T. Reza, N. Mehedi, N. A. Tasneem, and M. Ashraful Alam, "Identification of crop consuming insect pest from visual imagery using transfer learning and data augmentation on deep neural network," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–6.

[27] K. Xiong, Q. Li, Y. Meng, and Q. Li, "A study on weed detection based on improved yolo v5," in *2023 4th International Conference on Information Science and Education (ICISE-IE)*, 2023, pp. 1–4.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[29] P. Malik, M. K. Parida, and P. Ganaraj, "Insect localization and detection using object detection techniques," in *2023 9th International Conference on Signal Processing and Communication (ICSC)*, 2023, pp. 362–367.

[30] X. Wu, C. Zhan, Y. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *IEEE CVPR*, 2019, pp. 8787–8796.

[31] B. Shah and H. Bhavsar, "Time complexity in deep learning models," *Procedia Computer Science*, vol. 215, pp. 202–210, 2022, 4th International Conference on Innovative Data Communication Technology and Application. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922020944

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2015, pp. 3431–3440. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298965

[33] V. Wiley and T. Lucas, "Computer vision and image processing: A paper review," *International Journal of Artificial Intelligence Research*, vol. 2, p. 22, 02 2018.

[34] A. Rush, "Cnn feature map interpretation and key-point detection using statistics of activation layers," Ph.D. Thesis, Santa Clara University, 2023, engineering Ph.D. Theses, 51. [Online]. Available: https://scholarcommons.scu.edu/eng_phd_theses/51

[35] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[36] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," 2020.

[37] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219313803

[38] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222008426

[39] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2016.

[40] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:16489696

[41] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018, special issue on deep reinforcement learning. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608017302976

[42] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017.

[43] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digital Signal Processing*, vol. 132, p. 103812, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200422004298

[44] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[45] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[47] F. Joiya, "Object detection: Yolo vs faster r-cnn," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 04, no. 09, p. 1911, 2022, peer-Reviewed, Open Access, Fully Refereed International Journal. [Online]. Available: https://www.irjmets.com

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[49] B. Gong, D. Ergu, C. Ying, and B. Ma, "A method for wheat head detection based on yolov4," 09 2020.

[50] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, “Cspnet: A new backbone that can enhance learning capability of cnn,” 2019.

[51] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?” 2019.

[52] K. He, X. Zhang, S. Ren, and J. Sun, *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. Springer International Publishing, 2014, p. 346–361. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23

[53] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, 2023. [Online]. Available: http://dx.doi.org/10.3390/make5040083

[54] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-iou loss: Faster and better learning for bounding box regression,” 2019.

[55] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, “Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection,” 2020.

[56] P. Bindschaedler, K. Koerber, H. M. M. Bastiaans, J. Dietz, W. V. Deyn, J. Langewald, and F.-J. Braun, “Method for controlling pests of plants,” Patent, 2017.

[57] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[58] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019.

[59] P. Venkatasaichandrakanthand and M. Iyapparaja, “Pest detection and classification in peanut crops using cnn, mfo, and evita algorithms,” *IEEE Access*, vol. 11, pp. 54 045–54 057, 2023.