

Article



Computational Intelligence Approach for Fall Armyworm Control in Maize Crop

Alex B. Bertolla ^{1,2,*} and Paulo E. Cruvinel ^{1,2,*}

- ¹ Embrapa Instrumentation, São Carlos 13561-206, SP, Brazil
- ² Post Graduation Program in Computer Science, Federal University of São Carlos, São Carlos 13565-905, SP, Brazil
- * Correspondence: alex.bertolla@embrapa.br (A.B.B.); paulo.cruvinel@embrapa.br (P.E.C.)

Abstract: This paper presents a method for dynamic pattern recognition and classification of one dangerous caterpillar species to allow for its control in maize crops. The use of dynamic pattern recognition supports the identification of patterns in digital image data that change over time. In fact, identifying fall armyworms (*Spodoptera frugiperda*) is critical in maize production, i.e., in all of its growth stages. For such pest control, traditional agricultural practices are still dependent on human visual effort, resulting in significant losses and negative impacts on maize production, food security, and the economy. Such a developed method is based on the integration of digital image processing, multivariate statistics, and machine learning techniques. We used a supervised machine learning algorithm that classifies data by finding an optimal hyperplane that maximizes the distance between each class of caterpillar with different lengths in N-dimensional spaces. Results show the method's efficiency, effectiveness, and suitability to support decision making for this customized control context.

Keywords: image processing; pattern recognition; pests classification; machine learning; deep learning

1. Introduction

Maize (*Zea mays* L.) is one of the most important agricultural crops and one of the most cultivated cereals in the world. There are signs of maize cultivation dating back approximately seven thousand years in the regions where the country of Mexico's is located today. It was the main food source for the American people of that period [1,2].

According to the United States Department of Agriculture (USDA) [3], Brazil is the third largest maize producer in the world, with actual production of 122 million tons in an area of 21.5 million hectares in its 2023/2024 harvest and production forecasts of 127 million tons in an area of 22.30 million hectares for its 2024/2025 harvest. For comparison, the United States and China are ahead of Brazil in this regard, with forecast productions for their 2023/2024 harvests of 389.67 and 288.84 million tons, respectively.

A large range of pests and diseases attacks the maize crop during its different stages of plant development, which severely affects its productive potential [4]. Among other caterpillar pests, such as *Helicoverpa armígera*, *Helicoverpa zea*, and *Elasmopalpus lignosellus Zeller* [5], the fall armyworm (*Spodoptera frungiperda*) (FAW) is one of the most notorious and voracious, able to cause losses that could reach approximately 70% of production, as it attacks the plant still in its formation stage [6]. The extensive losses caused by this pest can heavily affect the economy, as it is also considered a serious pest for other important crops in the world, such as soybean, cotton, potato, etc. [7]. Table 1 presents the specific patterns of the above-cited caterpillars and their particularities.



Academic Editors: Sandra V.B. Jardim, Rolando Miragaia and José Carlos Bregieiro Ribeiro

Received: 14 January 2025 Revised: 27 March 2025 Accepted: 29 March 2025 Published: 3 April 2025

Citation: Bertolla, A.B.; Cruvinel, P.E. Computational Intelligence Approach for Fall Armyworm Control in Maize Crop. *Electronics* **2025**, *14*, 1449. https://doi.org/10.3390/ electronics14071449

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/ licenses/by/4.0/).

Type of CaterpillarSpecific Patterns		Color Patterns	Maximum Size (mm)
Inverted "Y" marking in the head area and four smaller dorsal spots in a trapeze arrangement on most Gr segments and in a square arrangement on the last segment		Greenish to dark brown	40 mm
Helicoverpa armígera Smooth saddle-shaped tubercles with hairs at the apex		Yellowish–white to reddish–brown or green	40 mm
Helicoverpa zea Stripes of other colors arrang the sides of the body		White and yellow, turning to brown	35 mm
Elasmopalpus lignosellusBrown, purple, or dark-brownZellertransverse streaks		Yellowish with red stripes; before hatching, it takes on a black coloration	22 mm

Table 1. Characteristics of caterpillars.

The first reports of FAW come from regions of North and South America, where it has been considered a constant pest [8]; however, in recent years, the presence of this pest has also been reported in different regions around the globe, including Asia [9], Africa [7], and Oceania [10]. In 2020, Kinkar et al. elaborated a technical report to the European Food Safety Authority (EFSA), where emergency measures were in place to prevent the introduction and spread of FAW within the European Union (EU). Due to the high spread capacity of adults, detection of moths at low levels of population is crucial to avoid further spread of this pest [11].

The developmental sequence of the FAW can be seen in Figure 1, which characterizes its different stages of growth, also called instars [5,12]. It is essential to emphasize that at about the neonate stage (newborn), given the size of the pest, it is impractical to attempt its detection by imaging; rather, it is more prudent to detect colonies (eggs).



Figure 1. Fall armyworm (Spodoptera Frugiperda) stages of growth [12].

However, according to agricultural pest experts, instars 5 and 6 can be regarded as a single instar based on their characteristics, method of control, and damage to the maize crop. Thus, the main focus of FAW pattern classification should be instars 1 to 5.

The current method of monitoring for FAW in maize includes trapping males using a pheromone odor similar to that of females. Once the pest is confirmed to be present in crop areas, insecticides are used for control.

However, such a technique can capture only the specimens that are already about to transform into moths or those that are already in that form, at which point they no longer cause significant damage to production [13]. Furthermore, traditional methods executed by humans are also labor-intensive and subjective, as they depend on human efforts [14].

This discrepancy between the current pest detection method and the intended result (to detect pests while they are still at their harmful stages) motivated this research study to find other methods for the early detection of this pest in cultured areas. Specifically, the focus of this study is on the development of a method for dynamic pattern recognition and classification for FAW based on the integration of digital image and signal processing, multivariate statistics, and machine learning (ML) techniques to favor the productivity of the maize crop.

The concept of computer vision seeks, through computer models, to reproduce the ability and functions of human vision, that is, the ability to see and interpret a scene. The ability to see can be implemented through the use of image acquisition devices and suitable methods for pattern recognition.

Nowadays, intelligent systems in agriculture that offer support for decisions in productive areas are equipped with capabilities based on machine learning processes. ML describes the capacity of systems to learn from customized problem-training data to automate and solve associated tasks [15–17]. In conjunction with such a concept, Deep learning (DL) is also a machine learning process but based on artificial neural networks with a set of specific-purpose layers [18,19]. The first layer of a neural network is the input layer, where the model receives input data. Such networks also include a convolutional layer, which uses filters to detect features in input data. In addition, to reduce the spatial dimensions of data, a pooling layer is used, which decreases the computational load. Likewise, after that, to allow for output, a fully connected layer is also included in such an arrangement, since each neuron in the layer is connected to every neuron in the previous and subsequent layer, which allows for analysis approaches [20].

Furthermore, ML and artificial intelligence (AI) can help in the decision-making process to establish a diagnosis with the aim of controlling this pest in a maize crop area. Currently, image and signal processing techniques are being utilized in several domains, most prominently in medicine [21], industry, security, and agriculture [22], among others.

Image acquisition has been verified to be a promising approach to the detection and identification of insect pests and plant diseases. In 2010, Sankaran et al. sought to identify diseased plants with greening or with nutritional deficiency through a method based on mid-infrared spectroscopy, where the samples are analyzed by a spectrometer [23]. In 2014, Miranda et al. proposed the study of different digital image processing techniques for the detection of pests in rice paddies by capturing images in the visible spectrum (RGB). They proposed a methodology in which images are scanned pixel by pixel both horizontally and vertically. This process is conducted in such a way that it is possible to detect and calculate the size (in pixels) of detected pests [24].

In 2015, Buades et al. presented a method for the filtering of digital images based on non-local means [25]. This method was compared with traditional methods of digital image filtering, such as Gaussian filtering, anisotropic diffusion filtering, and neighborhood-based filtering for white noise reduction. In 2011, Mythil and Kavitha compared the efficiencies of applying different types of filters to reduce noise in color digital images [26]. Mishra et al. compared Wiener, Lucy–Richardson, and regularized-filter digital filters to reduce noise in digital images [27].

In 2021, Bertolla and Cruvinel presented a method for the filtering of digital images degraded by non-stationary noise. In their research study, these authors added Gaussian-type noise with different levels of intensity to images of agricultural pests. Their approach allowed for the observation of images of maize pests with random noise signals for their [28] processing.

In 2013, He et al. highlighted methodologies based on image segmentation for the identification of pests and diseases in crops. With these methods, pests and diseases of cotton crops could be identified through segmentation techniques based on pseudo-colors (HSI and YCbCr), as well as in the visible spectrum (RGB) [29]. In 2015, Xia et al. detected small insect pests in low-resolution images that were segmented using the watershed method [30]. In 2017, Kumar et al. used the image segmentation technique known as adaptive thresholding for the detection and counting of insect pests. Such a method consists of computing the threshold of each pixel of the image by interpolating the results of the sub-images [31]. In 2018, Sriwastwa et al. compared color-based segmentation with Otsu segmentation and edge detection methods. Their experiments were initially performed with the Pyrilla pest (Pyrilla), found in sugarcane cultivation (Saccharum officinarum). Subsequently, the same methods were applied to images of termites (Isoptera) found in maize cultivation. For color-based segmentation, images were converted to the CIE L*a*b* [32] color space. For the feature extraction and pattern recognition of agricultural pests and diseases, in 2007, Huang applied an artificial neural network with the backpropagation algorithm for the classification of bacterial soft rot (BSR), bacterial brown spot (BBS), and Phytophthora black rot (PBR) in orchid leaves. The color and texture characteristics of the lesion area caused by the diseases were extracted using a co-occurrence matrix [33].

In 2011, Sette and Mailard proposed a method based on texture analysis of georeferenced images for the monitoring of a certain region of the Atlantic Forest based on images analyzed in the visible spectrum. Metrics such as contrast, entropy, correlation, inverse moment of difference, and second angular momentum were also extracted from the co-occurrence matrix [34].

On the other hand, machine learning-based approaches were discussed in 2008 by Ahmed et al., who then developed a real-time machine vision-based methodology for the recognition and control of invasive plants. The proposed system also had the objective of recognizing and classifying invasive plants into broad-leaf and narrow-leaf classes based on the measurement of plant density through masking operations [35].

In 2012, Guerrero et al. proposed a method based on a support vector machine (SVM) classifier for identification of weeds in maize plantations. For the classification process, they used SVM classifiers with a polynomial kernel, radial basis function (RBF), and sigmoid function [36].

In 2015, for the classification of different leaves, Lee et al. used a Convolutional Neural Network (CNN) based on the AlexNet neural network and a deconvolutional network to observe the transformation of leaf characteristics. For the detection of pests and diseases in tomato cultivation, a methodology based on machine learning techniques was proposed by Fuentes et al. [37]. Three models of neural networks were used to perform this task. To recognize pests and diseases (objects of interest) and their locations in the plant, faster region-based convolutional neural networks (F-CNNs) and region-based fully convolutional neural networks (R-FCNs) have been used [38].

In 2017, Thenmozhi and Reddy presented digital image processing techniques for insect detection in the early stage of sugar cane crops based on the extraction of nine geometrical features. The authors used the Bugwood image database for image sample composition [39]. In 2023, the same image database was used by Tian et al. The authors proposed a model based on the deep learning architecture to identify nine kinds of tomato

diseases [40]. In 2019, Evangelista proposed the classification of flies and mosquitoes based on the frequency of their wing beats. Fast Fourier transform and a Bayesian classifier were used [41].

In 2018, Nanda et al. proposed a method for detecting termites based on SVM classifiers, using pieces of wood previously divided into two classes: infested and not infested by termites. SVM classifiers with linear kernel, RBF, polynomial, and sigmoid functions were used on datasets obtained by microphones, which captured sound signals from termites [42]. In 2019, Lui et al. presented a methodology for extracting data from intermediate layers of CNNs with the purpose of using these data to train a classifier and, thus, make it more robust. Features extracted from the intermediate layers of a CNN were representative and could significantly improve the accuracy of the classifier. To test the effectiveness of that method, CNNs such as AlexNet, VggNet, and ResNet were used for feature extraction. The extracted features were used to train classifiers based on SVM, naive Bayes, linear discriminant analysis (LDA), and decision trees [43].

In 2019, Li et al. trained a maximum likelihood estimator for the classification of maize grains. "Normal" and "damaged" classes were defined, the latter having seven subclassifications [44]. In 2019, Abdelghafour et al. presented a framework for classifying the covers of vines in their different phenological stages, that is, foliage, peduncle, and fruit. For this task, a Bayesian classifier and a probabilistic maximum a posteriori (MAP) estimator were used [45].

In 2022, Moreno and Cruvinel presented results related to the control of weed species with instrumental improvements based on a computer vision system for direct precision spray control in agricultural crops for the identification of invasive plant families and their quantities [46].

In 2024, Wang and Luo proposed a method to identity specific pests that occur in maize crops. This method is based on a YOLOv7 network and the SPD-Conv module to replace the convolutional layer, i.e., to realize small target feature and location information. According to the authors, the experimental results showed that the improved YOLOv7 model was more efficient for such control [47].

In 2024, Liu et al. proposed a model to detect maize leaf diseases and pests. These authors presented a multi-scale inverted residual convolutional block to improve models' ability to locate the desired characteristics and to reduce interference from complex back-grounds. In addition, they used a multi-hop local-feature architecture to address problems regarding the extraction of features from images [48].

In 2025, Valderrama et al. presented an ML method to detect *Aleurothrixus floccosus* in citrus crops. This method is based on random sampling image acquisition, i.e., alternating the extraction of leaves from different trees. Techniques of imaging processing, including noise reduction, edge smoothing, and segmentation, were also applied. The final results were acceptable, and the authors used a dataset of 1200 digital images for validation [17].

Also in 2025, Zhong et al. proposed a flax pest and disease detection method for different crops based on an improved YOLOv8 model. The authors employed the Albumentations library for data augmentation and a Bidirectional Feature Pyramid Network (BiFPN) module. This arrangement was organized to replace the original feature extraction network, and the experimental results demonstrated that the improved model achieved significant detection performance on the flax pest and disease dataset [49].

This paper presents the integration of digital image processing, multivariate statistics, and computational intelligence techniques, focusing on a method for dynamic pattern recognition and classification of FAW caterpillars. Customized context analysis is also performed, taking into account ML and DL based on SVM classifiers and an AlexNet CNN (A-CNN) [50] through the use of the Tensorflow framework.

After this Introduction, the remainder of this paper is organized as follows: Section 2 introduces the materials and methods, Section 3 presents the results and respective discussions, and Section 4 provides conclusions and includes suggestions for continuity and future research.

2. Materials and Methods

All experiments were performed in Python (version 3.11), i.e., by using both the image processing and ML libraries in openCV, as well as the scikit-image and scikit-learn algorithms. We also considered an operating platform with a 64-bit CPU Intel (R) model Core(TM) i7-970, 16 Gb RAM, and Microsoft Windows 11 operating system. Figure 2 shows a block diagram of the method for classifying the patterns of FAW caterpillars.



Figure 2. Block diagram for classification of the classification of dynamic patterns of fall armyworm (*Spodoptera frugiperda*) in maize crop.

Concerning the dataset used for validation, the choice of images was based mainly on their quality and diversity. For this study, the Insect Images dataset, which is a subset of the Bugwood Image Database System, was selected. Currently, the Bugwood Image Database System is composed of more than 300 thousand images divided into more than 27 thousand subgroups. Another important factor for the use of this dataset is that most of the images were captured in the field, that is, they were influenced by lighting and have variations in scale and size, among other characteristics resulting from acquisition in a real environment. Therefore, in order to minimize the effects of lighting on the images, a set of digital images from leaves and cobs with the presence of FAW was taken into account, acquired under close lighting intensities, in addition to the inclusion of geometric feature extraction, together with color and texture information, for pattern recognition. Table 2 outlines the characteristics of the images used to validate the developed method.

Type of Archive	JPG/JPEG
Color space	RGB
Image width	3072 pixels
Image height	2048 pixels
Image resolution	72 ppi (pixels per inch)
Pixel size	$\approx 0.35 \text{ mm}$

Table 2. Characteristics of images of fall armyworm (Spodoptera frugiperda) in Insect Image dataset.

The restoration process considers the use of a degradation function (*H*), resulting in a restored image (f'(x, y)). When the degradation is exclusively due to noise, the *H* function is applied with a value equal to 1. On the other hand, a filter (g(x, y)) enables an image to be obtained such that it presents a better result in terms of the Signal-to-Noise Ratio (SNR). As a result, a restored image ($\hat{f}(x, y)$) is obtained. The process of restoring noisy images can be applied to both the spatial and frequency domains.

For noise filtering of the acquired images, the use of digital filters and the presence of noise, mainly random Gaussian and impulsive noises, were considered [51]. In digital image processing, noise can be defined as any change in the signal that causes degradation or loss of information from the original signal, which can be caused by lighting conditions of the scene or object, the temperature of the signal capture sensor during the acquisition of the image, or transmission of the image, among other factors [28].

For the problem regarding FAW, it has been observed that only noises present in the considered images need to be treated. Thus, *H* equals 1, and additive noises resulting from temperature variation of the image capture sensor and influenced by lighting conditions can be represented as follows:

$$\hat{f}(x,y) = \eta(x,y) + f(x,y)$$
, (1)

where f(x, y) represents the original image, $\eta(x, y)$ is the noise added to the original image, and $\hat{f}(x, y)$ is the noisy image [26,52].

In this work, image restoration was performed in the spatial domain. The use of Gaussian filters [53] and non-local means [25] was evaluated. The application of a Gaussian filter has the effect of smoothing an image; the degree of smoothing is controlled by the standard deviation (σ). Its kernel follows the mathematical model expressed as follows:

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} ,$$
 (2)

where *x* and *y* represent the kernel dimensions of the filter and σ is the value of the standard deviation of the Gaussian function with $\sigma > 0$.

The application of a non-local mean (NLM) filter is based, as the name suggests, on non-local mean measures. The NLM filter searches for the estimated value of the intensity of each pixel (*i*) in a certain region of the image (*f*), then calculates the weighted average of this region. The similarity is estimated according to an image with noise (*g*) in the form of $g(i)|i \in f, NLM_{[g](s)}$, where *NLM* represents the estimated value of a given pixel (*i*) [54] as follows:

$$NLM_{[g](s)} = \frac{1}{C_{(i)}} \sum_{j \in f} \omega(i, j) g(j) \quad ,$$
(3)

where $C_{(i)}$ is a normalizing factor, i.e., $C_{(i)} \neq 0$, and $\omega(i, j)$ represents the similarity of the weights of pixels *i* and *j* to satisfy the conditions of $0 \le \omega(i, j) \le 1$ and $\sum_{j \in f} \omega(i, j) = 1$. In addition, the weight of $\omega(i, j)$ [55] is calculated as follows:

$$\omega_{(i,j)} = \frac{1}{C_{(i)}} \exp^{-\frac{|(N_i)^2 - (N_j)^2|}{\sigma^2}} , \qquad (4)$$

where N_i and N_j are vectors of pixels whose values are related not only to similarity measures but also to the Gaussian weighted Euclidean distance at which intensities of the gray levels are within a square neighborhood centered at positions *i* and *j*, respectively.

For color-space operations, the use of the HSV and CEI L*a*b* color spaces was evaluated based on images acquired in the RGB color space [56].

In reference to digital image processing, the RGB color space applies mainly to the image acquisition and result visualization stages. However, because of its low capacity to capture intensity variation in color components [57], its use is not recommended in the other stages of the process.

To convert an RGB image to the HSV color space, because of the more natural representation of RGB colors for human perception, it is necessary to normalize the values of the R, G, and B components, measuring their respective maximum and minimum values, then convert them to the HSV color space as follows:

$$H = \begin{cases} 60^{\circ} - \left(\frac{G' - B'}{\Delta} \mod 6\right) & \text{if } C_{max} = R' \\ 60^{\circ} - \left(\frac{B' - R'}{\Delta} + 2\right) & \text{if } C_{max} = G' \\ 60^{\circ} - \left(\frac{R' - G'}{\Delta} + 4\right) & \text{if } C_{max} = B' \end{cases}$$
(5)

$$S = \begin{cases} 0 & \text{if } \Delta = 0\\ \frac{\Delta}{C_{max}} & \text{if } \Delta \neq 0 \end{cases}$$
(6)

$$V = C_{max} \quad , \tag{7}$$

where C_{max} is the maximum value of the *R*, *G*, and *B* components; C_{min} is the minimum value of the *R*, *G*, and *B* components; and *H*, *S*, and *V* are the components of the HSV color space.

It is also possible to convert images from the HSV to the RGB color space, considering the following:

$$(R_1, G_1, B_1) = \begin{cases} (C, X, 0) & \text{if} & 0^\circ \le H < 60^\circ \\ (X, C, 0) & \text{if} & 60^\circ \le H < 120^\circ \\ (0, C, X) & \text{if} & 120^\circ \le H < 180^\circ \\ (0, X, C) & \text{if} & 180^\circ \le H < 240^\circ & \prime \\ (X, 0, C) & \text{if} & 240^\circ \le H < 300^\circ \\ (C, 0, X) & \text{if} & 300^\circ \le H < 360^\circ \end{cases}$$
(8)

where R_1 , G_1 , and B_1 represent points on the faces of the RGB cube, whereas *C* represents the chroma component.

On the other hand, the conversion from the RGB to the CIE L*a*b* color space follows the method described below. The *Commission Internationale de l'Eclairage* (CIE), or the International Commission on Illumination, defines the sensation of color based on the elements of luminosity, hue, and chromaticity. Thus, the condition of existence of color is based on three elements: illuminant, object, and observer [58]. Accordingly, the color space known as CIE Lab started to consider the "L" component as a representation of luminosity, ranging from 0 to 100; the "a" component as a representation of chromaticity, ranging from green (negative values) to red (or magenta for positive values); and the "b" component also as a representation of chromaticity, varying from yellow (for negative values) to blue (or cyan for positive values) [59]. In 1976, the CIE L*a*b* standard was created based on improvements of the CIE Lab created in 1964. The new standard provides more accurate color differentiation concerning human perception [60]. However, given that the old standard is still widely used, the use of asterisks (*) was adopted in the nomenclature of the new standard.

Because the CIE L*a*b* color space standard, like its predecessor, is based on the CIE XYZ color standard, conversion from RGB images occurs in two steps [61]. First, the RGB image is converted to the CIE XYZ standard, that is,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} .$$
(9)

Once the image has been converted to the CIE XYZ standard, it can be converted to the CIE L*a*b* standard, considering the following:

$$L^{*} = \begin{cases} 116 \left(\frac{Y}{Y_{n}}\right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_{n}} > 0.008856 \\ 903, 3 \left(\frac{Y}{Y_{n}}\right) & \text{if } \frac{Y}{Y_{n}} \le 0.008856 \end{cases}$$
(10)

$$a^* = 500 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} \right] , \qquad (11)$$

$$b^* = 200 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n} \right)^{\frac{1}{3}} \right]$$
 (12)

The segmentation step aims to divide, or isolate, regions of an image, which could be labeled as foreground and background objects. Thus, foreground objects are called regions of interest (ROIs) of the image, that is, regions where patterns related to the end objective are sought for the identification of FAW. Background objects are any other objects that are not of interest [62].

Segmentation algorithms are generally based on two basic properties: discontinuity, such as edge detection and the identification of borders between regions, and similarity, which is the case of pixel allocation in a given region [63].

In binary images, the representation of pixels with values of 0 (black) normally indicates the background of the image, whereas that of pixels with values of 1 (white) indicates the object(s) of interest [64].

A binary image (b(x, y)) is generated with the application of a threshold (*T*) to the histogram of the original image (f(x, y)), considering the following:

$$b(x,y) = \begin{cases} 0, & \text{if } f(x,y) < T \\ 1, & \text{if } f(x,y) \ge T \end{cases}$$
(13)

The threshold value can be chosen through a manual analysis from the histogram of an image or the use of an automatic threshold selection algorithm. In this case, seed pixels were used. Additionally, the use of Otsu's method was considered. This method performs non-parametric and unsupervised discriminant analysis and automatically selects the optimal threshold based on the intensity values of the pixels of a digital image, allowing for a better separation of classes [65].

In a 2D digital image with dimensions of $N \times M$ and an intensity level of L, where n_i denotes the number of pixels of intensity i, and MN is the total number of pixels, the histogram is normalized considering the following [66] :

$$p_i = \frac{n_i}{MN} \quad , \tag{14}$$

where $p_i \ge 0$ and $\sum_{i=1}^{L} p_i = 1$.

The operation of a threshold (T(k) = k, where 0 < k < L - 1) divides the *L* intensity levels of an image into two classes (C_1 and C_2 , representing the object of interest and the background of the image, respectively, where C_1 consists of all pixels in the range of [0, k] and C_2 covers the range of [k + 1, L - 1]). This operation is defined as follows:

$$P_1(k) = \sum_{i=0}^{k} p_i , \qquad (15)$$

where $P_1(k)$ is the probability of a pixel (*k*) being assigned to class C_1 .

Likewise, the probability of occurrence of class C_2 is expressed as follows:

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \quad . \tag{16}$$

The values of the average intensities of classes C_1 and C_2 are given considering the following equations:

$$m_1(k) = \sum_{i=0}^k iP(i|C_1) = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i \quad , \tag{17}$$

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i|C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \quad .$$
(18)

The average cumulative intensity (m(k)) is expressed as follows:

$$m(k) = \sum_{i=0}^{k} i p_i$$
 (19)

The optimal threshold can be obtained via the minimization of one of the discriminant functions, described as follows:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2} , \ \eta = \frac{\sigma_B^2}{\sigma_G^2} , \ \kappa = \frac{\sigma_T^2}{\sigma_W^2} ,$$
 (20)

where σ_W^2 is within-class variance, σ_B^2 is cross-class variance, and σ_G^2 is global variance, respectively expressed as follows:

$$\sigma_B^2 = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1 - m)^2}{P_1 (1 - P_1)} \quad , \tag{21}$$

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 pi \quad . \tag{22}$$

The greater the difference in the average values of m_1 and m_2 , the greater the variance between classes (σ_B^2), confirming it to be a separability measure [66]. Likewise, if σ_G^2 is a constant, it is possible to verify that η is also a measure of separability and that maximizing this metric is equivalent to maximizing σ_B^2 . Thus, the objective is to determine the value of k to maximize the variance between classes. Therefore, the threshold (k) that maximizes the η function is selected based on the following:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2} \quad . \tag{23}$$

Also, when the optimal threshold (k^*) is obtained, the original image (f(x, y)) is segmented considering the following:

$$b(x,y) = \begin{cases} 0, & \text{if } f(x,y) \le k^* \\ 1, & \text{if } f(x,y) > k^* \end{cases}$$
(24)

The use of Otsu's method automates the process of segmenting images containing objects that represent FAW both in maize leaves and cobs. Additionally, after the segmentation method was applied, the extraction of features from these patterns, through the use of the methods of the histogram of oriented gradient (HOG) [67] and invariant moments of Hu [68], was considered.

Herein, the HOG descriptor is applied in five stages [69]. The first involves transforming the segmented image in the CIE L*a*b* color space to grayscale (in 8-bit or 256-tone conversion), whereas the other steps involve calculating the intensities of the gradients; grouping the pixels of the image into cells; grouping these cells into blocks; and, finally, extracting characteristics of the magnitude of the gradient, according to the following equation:

$$m(x,y) = \sqrt{[f_u(x,y)]^2 + [f_v(x,y)]^2} \quad , \tag{25}$$

where *m* is the magnitude of the feature vector at point (x, y). Additionally, $f_u(x, y)$ consists of a component in *u* directions, and $f_v(x, y)$ is a component in *v* directions . It is possible to obtain the direction of the gradient vector $(\theta(x, y))$ in the following form:

$$\theta(x,y) = \tan^{-1} \frac{f_v(x,y)}{f_u(x,y)} .$$
(26)

In addition, for geometrical feature extraction purposes, the Hu invariant moments descriptor was considered [70]. First, it is necessary to calculate the two-dimensional moments. They can be defined as polynomial functions projected in a 2D image (f(x, y)) with dimensions of $M \times N$ and order (p + q).

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad , \tag{27}$$

where γ is defined as

$$\gamma = \frac{p+q}{2} + 1 \quad , \tag{28}$$

for $p + q = 2, 3, \ldots$, positive integers $\in \mathbb{Z}$.

Then, the invariant moments can be calculated as follows:

$$\phi_1 = \eta_{20} + \eta_{02} \quad , \tag{29}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad , \tag{30}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad , \tag{31}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad , \tag{32}$$

$$\phi_{5} = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^{2} - 3(\eta_{21} + \eta_{03})^{2}] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) ,$$

$$(33)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) ,$$
(34)

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] .$$
(35)

Furthermore, in this work, we used Principal Component Analysis (PCA) [71] to reduce the dimensionality of the vector. We consider a data array (\mathbf{X}) with *n* observations and *m* independent variables.

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} .$$
(36)

The principal components can be measured as a set of *m* variables ($X_1, X_2, ..., X_m$) with means ($\mu_1, \mu_2, ..., \mu_m$) and variance ($\sigma_1^2, \sigma_2^2, ..., \sigma_m^2$), in which covariance between the *n*-th and *m*-th variables takes the following form:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1m}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{n1}^2 & \cdots & \sigma_{nm}^2 \end{bmatrix} , \qquad (37)$$

where Σ is the covariance matrix. Eigenvalues and eigenvectors are measured ((λ_1, e_1), (λ_2, e_2), . . . , (λ_m, e_m), where $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_m$) and associated with Σ , where the *i*-th principal component is defined as follows:

$$Z_i = e_{i1}X_1 + e_{i2}X_2 + \ldots + e_{im}X_m , \qquad (38)$$

where Z_i is the *i*th principal component. The objective is to maximize the variance of Z_i as follows:

$$Var(Z_i) = Var(e'_i \mathbf{X}) = e'_i Var(\mathbf{X})e_i = e'_i \Sigma e_i \quad , \tag{39}$$

where i = 1, ..., m. Thus, the spectral decomposition of the matrix (Σ) is expressed as $\Sigma = \mathbf{P} \mathbf{\Lambda} \mathbf{P}'$, where **P** is the composite matrix according to the eigenvectors of Σ and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of Σ . Thus,

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix} .$$
(40)

The principal component of greatest importance is defined as the one with the greatest variance that explains the maximum variability in the data vector, as the second highest variance represents the second most important component, and so on, to the least important component.

The vector of reduced dimensionality features is composed of normalized eigenvectors, representing the descriptors of the FAW descriptors in the images. The feature vector comprises the input data for pattern recognition, involving ML.

As mentioned previously, in other words, ML may also be understood as the ability of a computational system to improve performance in a task based on experience [72]. In this work, the classification technique is related to ML—specifically, supervised learning [73]. Supervised learning is based on existent and classified patterns serving as training examples that enable a classifier to be efficiently generalized to new datasets [18]. In this context, the feature vector, with reduced dimensionality, is used for classification according to its position in the feature space.

In this work, after having defined the feature vector, as a next step, the application of computational intelligence is considered, i.e., based not only on SVM classifiers [74] but also taking into account the A-CNN through the use of the Tensorflow framework [19,75] for evaluation. In such a way, it is possible to carry out a context evaluation to choose between the use of ML or DL models in such a customized pest control problem.

SVM classifiers were selected for use in this study, since they have been quite well used for the classification of agricultural data, as reported in [76]. SVMs can be established based on linear behavior or even non-linear behavior. In this work, two types of kernel that are associated with these functionalities were evaluated to determine which one leads to the best operability, accuracy, precision, and other parameters associated with the analysis of the behaviors of these classifiers [77].

Classifiers with linear behavior use a hyperplane that maximizes the separation between two classes from a training dataset (**x**) with *n* objects ($x_i \in \mathbf{X}$) and their respective labels ($y_i \in \mathbf{Y}$) such that **X** represents the input dataset and $\mathbf{Y} = (-1, +1)$ represents possible classes [72]. In this case, the hyperplane is defined as follows:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \ge 0, \quad \forall (\mathbf{x}_i, y_i) \in \mathbf{X}$$
, (41)

where **w** is the normal vector to the hyperplane, $\mathbf{w} \cdot \mathbf{x}$ is the dot product of vectors **w** and \mathbf{x} , and b is a fit term.

The maximization of the data separation margin in relation to $\mathbf{w} \cdot \mathbf{x} + b = 0$ can be obtained via the minimization of $\|\mathbf{w}\|$ [78]. The minimization problem is quadratic because the object function is quadratic and the constraints are linear. This problem can be solved via the introduction of the Lagrange function [79]. The Lagrange function must be minimized concerning \mathbf{w} and b, implying the maximization of the variables (α_i). The value of L is derived concerning \mathbf{w} and b.

This formulation is referred to as the dual form, whereas the original problem is referred to as the primal form. The dual form presents simpler restrictions and allows for Linearly separable datasets are classified efficiently by linear SVMs, with some error tolerance in the case of a linear SVM with smooth margins. However, in several cases, it is not possible to efficiently classify training data using this modality of a hyperplane [72], requiring the use of interpolation functions that allow for operation in larger spaces, that is, using non-linear SVM classifiers.

In that manner, it is possible for SVMs to deal with non-linear problems through a Φ function, mapping the dataset from its original space (input space) to a larger space (input space characteristics) [80], characterizing a non-linear SVM classifier.

Thus, based on the choice of Φ , the training dataset (**x**), in its input space (R^2), is scaled to the feature space (R^3) as follows:

$$\mathbf{\Phi}(\mathbf{x}) = \Phi(x_1, x_2) = (x^2, \sqrt[2]{2}x_1 x_2, x_2^2) \quad , \tag{42}$$

$$h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}) + b = w_1 x_1^2 + w_2 \sqrt[2]{2} x_1 x_2 + w_3 x_2^2 + b = 0 \quad .$$
(43)

In this way, the data are initially mapped to a larger space; then, a linear SVM is applied over the new space. A hyperplane is then found with a greater margin of separation, ensuring better generalization [81].

Given that the feature space can be in a very high dimension, the calculation of Φ might be extremely costly or even unfeasible. However, the only necessary information about the mapping is the calculation of the scalar products between the data in the feature space obtained through function kernels [72]. Table 3 presents the kernels selected to validate the developed method.

Kernel	Function $K(x_i, x_j)$	Parameters
Polynomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$	<i>δ</i> , <i>κ</i> e <i>d</i>
Radial basis function (RBF) kernel	$exp(-\sigma \ \mathbf{x}_i - \mathbf{x}_j\ ^2)$	σ
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)$	δ е κ

Table 3. Selected kernels to validate the developed method.

For each type of kernel, one should define a set of parameters, which must be customized as a function of the problem to be solved.

In addition, the use of a CNN based on a tensor is evaluated, which can be defined as a mathematical entity used for multidimensional representation of data. Its order determines the indices required for the component's access. In fact, tensors can be scalar, vectors, matrices, or even higher-order entries with *N* dimensions [82], as follows:

$$\mathbf{X} \in \mathbf{R}^{I_1 \times I_2 \times \dots I_N} \quad , \tag{44}$$

where I_k for $1 \le k \le N$ is the dimension of the k-th mode of the tensor and $x_{i_1,i_2,...,i_N}$ denote each element of the tensor (**X**).

A Tensor Network (TN) is defined as a collection of tensors that can be multiplied or compacted according to a predefined topology. In such an arrangement, the configuration can be obtained by taking into account two types of indices. One of them considers a linked index, which can connect tensors two at a time to structure the network arrangement. The other uses an open index, which makes it possible to connect tensors directly to the network being structured. Larger scale dimensions can also be obtained by using addition and multiplication operations and considering linked and open operators, respectively.

In this work, all parameters are considered, not only for the SVM classifiers but also for the A-CNN, to allow for control of FAW caterpillars in maize crops.

3. Results and Discussions

For this study, an image dataset of the FAW in maize crop was organized, with a total of 2280 images representing its five stages of development, that is, 456 images generated for each stage of development. Figure 3 shows the results of the image acquisition step.



Figure 3. Different stages of the Fall Armyworm (*Spodoptera frugperda*) found on maize leaves (**a**,**b**) and cobs (**c**,**d**), i.e., instar 1, instar 2, instar 3, and instar 4.

Based on the mean squared error (MSEs) and peak signal-to-noise ratios (PSNRs) of the images restored by the spatial filtering process, it was observed that the NLM filter yielded a better result than the Gaussian filter, as shown in Figure 4.



Figure 4. Evaluation of the (**a**) Mean Squared Error (MSE), and (**b**) Peak Signal-to-Noise Ration (PSNR) for the non-local mean filter validation.

Table 4 presents the parameters used for the application of the NLM filter. These parameters were obtained after several tests performed with this filter based on the available literature.

Table 4. Non-local mean filter parameters and values.

Parameter	Value	
Kernel	7×7	
Patch width	11	
Patch height	11	

The use of a kernel with dimensions of 7×7 pixels and a height and distance patch of 11 pixels allowed for the maximum attenuation of medium- and low-frequency noise effects, as well as the maintenance of the textural characteristics of the images.

For the image segmentation stage, tests were conducted using Otsu's method based on the conversion of the HSV and CIE L*a*b* color spaces. However, because of the verified restrictions for the H map of the HSV color space, it was decided that the CIE L*a*b* would be used instead to obtain an ideal segmentation process.

Figures 5–7 illustrate the image segmentation process using Otsu's method based on components a* and b*, where each image shows only one FAW found on the considered region of maize leaves.

Figures 8–10 illustrate the image segmentation process using Otsu's method and based on components a* and b*, where each image showed two FAWs found on the considered region of maize leaves.



Figure 5. Segmentation process using Otsu's method. (**a**) Original image, (**b**) channel a*, (**c**) channel a* histogram, and (**d**) segmented image result.



Figure 6. Segmentation process using Otsu's method. (a) Original image, (b) channel b*, (c) channel b* histogram, and (d) segmented image result.



Figure 7. Segmentation process using Otsu's method. (a) Original image, (b) segmented image by channel a*, (c) segmented image by channel b*, and (d) segmented image result.



Figure 8. Segmentation process using Otsu's method. (a) Original image, (b) channel a*, (c) channel a* histogram, and (d) segmented image result.



Figure 9. Segmentation process using Otsu's method. (a) Original image, (b) channel b*, (c) channel b* histogram, and (d) segmented image result.



Figure 10. Segmentation process using Otsu's method. (a) Original image, (b) segmented image by channel a^{*}, (c) segmented image by channel b^{*}, and (d) segmented image result.

When the histograms of the a* components of Figures 5 and 8 are analyzed, it can be seen that the pixels with the lowest values, represented by the blue color, refer to the pixels of maize plant leaves. Conversely, the highest-value pixels, in red, represent pest pixels and other anomalies present on the leaf. Therefore, to segment the pest from the rest of the image, only pixels with values above the threshold obtained by Otsu's method are considered.

However, based on the tests conducted on the segmentation of images of FAWs on leaves using Otsu's method and only the a* component of the CIE L*a*b* color space, it was determined that, despite the proven efficiency of this segmentation method, there is an evident need for a second segmentation stage. Assays were then performed using the b* component.

In histogram analysis of Figures 6 and 9, it can be seen that the pixels with the lowest values represent the pixels of the pest and some parts of the leaf. Therefore, to segment the pest from the rest of the image, only pixels with values below the threshold obtained by Otsu's method were considered.

Approximating the segmentation process using only the a* component, segmentation using the b* component also resulted in an image with parts of the leaf still in it. However, it was possible to verify that, in the spatial domain of the image, the non-segmented parts referring to the leaf in the segmentation with the a* component did not belong to the same spatial locations as the segmentation result achieved by the b* component. Therefore, based on the results obtained via the segmentation of the pest from the image on a leaf by the a* and b* components of the CIE L*a*b* color space, a new segmentation step was performed, this time in the form of an intersection. It can be observed from the results that segmentation by intersection proved to be efficient in the segmentation of images of pests on leaves.

Figures 11 and 12 illustrate the image segmentation process using Otsu's method and based on the b* components of FAWs found in the considered region of maize cobs.



In Figure 11, only one FAW caterpillar is shown. In Figure 12, two FAW caterpillars in different stages are shown.

Figure 11. Segmentation process using Otsu's method. (a) Original image, (b) channel b*, (c) channel b* histogram, and (d) segmented image result.



Figure 12. Segmentation process using Otsu's method. (a) Original image, (b) channel b*, (c) channel b* histogram, and (d) segmented image result.

For the segmentation of the FAW on maize cobs using Otsu's method, the results of the tests performed on the a* component of the CIE L*a*b* color space are shown to be inefficient, given the conversion process from the RGB color space. Thus, for the segmentation of pests on maize cobs, only the b* component was used.

The histograms of the map of the b* component (Figures 11 and 12) show that the highest-value pixels represent the maize cob pixels in the b* component. Thus, to segment the FAW caterpillars from all the collected images, pixels with values below the threshold obtained by Otsu's method were considered.

The results of the segmentation process showed that parts of the cob were not completely segmented. These results were expected for the segmentation of both images with spikes and images containing leaves because during the tests performed to validate the segmentation step of the standard method, the complexity of image formation was verified in terms of the two values of the pixels that constitute both the FAW and the background of the image.

Thus, it could be verified that for images of FAW caterpillars found on leaves, the segmentation process using Otsu's method and the a* and b* maps achieved a result considered ideal, whereas for images of FAW caterpillars found on cobs, only the segmentation using the map of the b* component was sufficient.

In relation to image descriptors, in this work, we considered the use of the HOG and Hu methodologies.

The HOG descriptor was used to extract texture features of the FAW. Table 5 displays the parameterization of the HOG descriptor.

Parameter	Value	
Pixels per cell	16 imes 16	
Cells per block	2 imes 2	
Numbers of orientations	9	
Feature vector	TRUE	
Transform SQRT	FALSE	

 Table 5. HOG descriptor parameters.

For the execution of the HOG descriptor, previously segmented images were resized to obtain a spatial resolution of 256×256 pixels. Once the parameters of the HOG descriptor were applied to the resized images, it was possible to generate a feature vector of 8100 positions in the form of $V_{[HOG]}$ for each image of the FAW, as illustrated in Figure 13.



Figure 13. HOG feature descriptor. (a) Segmented fall armyworm image; (b) HOG image.

Once the feature vector ($V_{[HOG]}$) was obtained through the use of the HOG descriptor, the Hu invariant moment descriptor was then applied to the FAW images, as demonstrated in Figure 14.



Figure 14. Hu invariant moment results. (a) Fall armyworm stage, 1 ($M_1 = [2.581]$, $M_2 = [5.231]$, $M_3 = [9.430]$, $M_4 = [10.169]$, $M_5 = [20.381]$, $M_6 = [-13.881]$, $M_7 = [-20.003]$). (b) Fall armyworm, stage 3 ($M_1 = [2.4329]$, $M_2 = [4.9603]$, $M_3 = [8.2253]$, $M_4 = [9.0384]$, $M_5 = [-18.623]$, $M_6 = [-11.739]$, $M_7 = [17.673]$).

Thus, for each image of the FAW, a feature vector $(V_{[Hu]})$ was generated, containing the seven invariant moments of Hu, that is, the shape and size features of the pests: M₁, M₂, M₃, M₄, M₅, M₆, and M₇.

Considering the obtained descriptors, one referring to texture characteristics (HOG) and another referring to geometric characteristics (Hu), it was possible to classify the patterns of the FAW in its different stages of development after applying PCA to reduce the feature vector.

In fact, in feature extraction using the HOG descriptor and the Hu invariant moment descriptor, the feature vectors ($V_{[HOG]}$ and $V_{[Hu]}$) were concatenated to generate a single vector of features ($V_{[HOG,Hu]}$) with 8107 positions, as illustrated in Figure 15.



Figure 15. Feature vector example (*Vetor*_[HOG,Hu]).

Therefore, because the values referring to texture, shape, and size characteristics are in different scales, it was necessary to normalize them before the generation of a database with the characteristic features of the patterns presented in each analyzed image.

Furthermore, by using PCA, it was possible to achieve a dimensionality reduction from 2280 to 128 principal components, maintaining approximately 98% of the variability of the original data, as shown in Figure 16.



Figure 16. Feature vector example (*Vetor*_[HOG,Hu]) obtained via PCA.

In the tests on the classification and ML stage, SVM classifiers with a linear kernel function and a sigmoid kernel function were considered. For the validation of each SVM classifier, both the accuracy and precision in classifying the FAW target stage were taken into account.

For the training and testing stages of the SVM classifiers and the CNN, dataset proportions of 50%:50%, 70%:30%, and 80%:20% were evaluated for training and testing, respectively. Tables 6, 7, and 8 present the results of SVM classifiers with a linear kernel

#4

#5

0.85

0.85

function and dataset proportions of 50% for training and testing, 70% for training and 30% for testing, and 80% and 20% for training and testing, respectively.

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
#1	0.81	0.63	0.71	183	0.59
# 2	0.83	0.69	0.75	183	0.64
#3	0.73	0.79	0.83	183	0.73

0.79

0.72

Table 6. Fall armyworm stage classification with SVM linear function kernel and dataset proportion of 50% for training and testing.

Table 7. Fall armyworm stage classification with SVM linear function kernel and dataset proportions of 70% for training and 30% for testing.

0.82

0.78

183

183

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
# 1	0.74	0.68	0.71	106	0.56
# 2	0.78	0.65	0.71	106	0.57
#3	0.82	0.79	0.80	106	0.70
#4	0.81	0.78	0.80	106	0.69
# 5	0.86	0.75	0.80	106	0.71

Table 8. Fall armyworm stage classification with SVM linear function kernel and dataset proportions of 80% for training and 20% for testing.

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
#1	0.71	0.61	0.66	72	0.50
# 2	0.72	0.65	0.69	72	0.53
#3	0.79	0.81	0.80	72	0.68
#4	0.81	0.78	0.79	72	0.68
# 5	0.82	0.75	0.78	72	0.67

Tables 9, 10, and 11 present the results of the SVM classifiers with sigmoidal kernel functions and dataset proportions of 50%:50%, 70%:30%, and 80%:20% for training and testing, respectively.

Table 9. Fall armyworm stage classification with SVM sigmoidal function kernel and dataset proportion of 50% for training and testing.

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
#1	0.80	0.87	0.83	183	0.72
# 2	0.80	0.81	0.81	183	0.68
#3	0.80	1.00	0.89	183	0.80
#4	0.81	0.99	0.89	183	0.80
# 5	0.80	0.85	0.82	183	0.71

0.72

0.68

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
#1	0.75	0.86	0.86	106	0.67
#2	0.77	0.83	0.80	106	0.68
#3	0.77	1.00	0.87	106	0.77
#4	0.77	0.99	0.87	106	0.77
# 5	0.75	0.87	0.81	106	0.89

Table 10. Fall armyworm stage classification with SVM sigmoidal function kernel and dataset proportions of 70% for training and 30% for testing.

Table 11. Fall armyworm stage classification with SVM sigmoidal function kernel and dataset proportions of 80% for training and 20% for testing.

SMV Classifier	Precision	Recall	F1 Score	Support Vectors	Accuracy (%)
#1	0.77	0.88	0.82	72	0.70
# 2	0.80	0.86	0.81	72	0.69
#3	0.78	1.00	0.88	72	0.78
#4	0.78	0.97	0.86	72	0.76
#5	0.77	0.86	0.81	72	0.68

Taking into account the classifiers having linear and sigmoidal function kernels, the assessed results revealed their efficiency in the dynamic classification of the FAW. It was possible to observe that the best results were obtained by using the sigmoidal function kernel. Therefore, a deeper evaluation of the SVM classifier based on such a function kernel was considered as follows.

For stage 1, the SVM classifier with a proportion 50%:50% of the dataset for training and testing presented the best result, with an accuracy rate of 72% and a precision rate of 80%. For stage 2, the SVM classifier with a proportion of 80%:20% of the dataset for training and testing, respectively, showed the best results based on precision and accuracy of 80% and 69%, respectively. For stage 3, the SVM classifier with a proportion of 50% of the dataset for training and testing showed the best result, with 80% accuracy and precision. For stage 4, the best result was demonstrated by the SVM classifier with a proportion of 50% of the dataset for training and testing. Finally, for stage 5, the SVM classifier that presented the best result was also the classifier with a proportion of 50% of the dataset for training and testing. Finally, for stage 5, the SVM classifier that presented the best result was also the classifier with a proportion of 50% of training and testing, resulting in 71% accuracy and 80% precision.

Figures 17–19 illustrate the confusion matrices and ROC curves for the performance of Classifier #1.

Based on the measurements of the false-positive rate and true-positive rate, the AUC measures resulting from each version of Classifier #1 could be analyzed. In this way, it could be verified that the proportion of 50% of the dataset used for training and testing led to the best result in relation to the classifiers set for this work placement, i.e., AUC = 52%, followed by the classification with a proportion of 70%:30% of the dataset used for training and testing, respectively, with AUC = 48%. The classification with a proportion of 80%:20% of the dataset used for training and testing, respectively, led to the result of AUC = 45%.

Figures 20–22 illustrate the confusion matrices and ROC curves for Classifier #2.

For Classifier #2, for the produced AUC measures, it was observed that the classifier with a proportion of 50% of the dataset used for training and testing obtained the best result in relation to the classifiers set for this work placement, i.e., AUC = 50%, followed by the classification with a proportion of 80%:20% of the dataset used for training and testing, respectively, with AUC = 49%. The classification with a proportion of 70%:30% of the dataset used for training and testing, respectively, obtained a result of AUC = 45%.



Figure 17. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #1 with sigmoidal function kernel and dataset proportion of 50% for training and 50% for testing.







Figure 19. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #1 with sigmoidal function kernel and dataset proportion of 80% for training and 20% for testing.



Figure 20. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #2 with sigmoidal function kernel and dataset proportion of 50% for training and 50% for testing.



Figure 21. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #2 with sigmoidal function kernel and dataset proportion of 70% for training and 30% for testing.



Figure 22. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #2 with sigmoidal function kernel and dataset proportion of 80% for training and 20% for testing.

Figures 23–25 illustrate the confusion matrices and ROC curves for Classifier #3.

For Classifier #3, for the produced AUC measures, it was observed that the classifier with a proportion of 50% of the dataset used for training and testing presented a result of AUC = 48%, and the same result was achieved for classification with a proportion of

70%:30% of the dataset used for training and testing, respectively. The classification with a proportion of 80%:20% of the dataset used for training and testing, respectively, achieved a result of AUC = 44%.



Figure 23. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #3 with sigmoidal function kernel and dataset proportion of 50% for training and 50% for testing.







Figure 25. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #3 with sigmoidal function kernel and dataset proportion of 80% for training and 20% for testing.

Figures 26–28 illustrate the confusion matrices and ROC curves for Classifier #4.



Figure 26. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #4 with sigmoidal function kernel and dataset proportion of 50% for training and 50% for testing.



Figure 27. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #4 with sigmoidal function kernel and dataset proportion of 70% for training and 30% for testing.



Figure 28. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #4 with sigmoidal function kernel and dataset proportion of 80% for training and 20% for testing.

With regard to Classifier #4, the classifiers with a proportion of 50% of the dataset used for training and testing presented the best result in relation to the classifiers set for this work placement, i.e., AUC = 49%, followed by the classification with a proportion of 70%:30% of the dataset used for training and testing, respectively, with AUC = 43%. The

classification with a proportion of 80%:20% of the dataset used for training and testing, respectively, achieved a result of AUC = 35%.

Figures 29–31 illustrate the confusion matrices and ROC curves for Classifier #5.



Figure 29. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #5 with sigmoidal function kernel and dataset proportion of 50% for training and 50% for testing.



Figure 30. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #5 with sigmoidal function kernel and dataset proportion of 70% for training and 30% for testing.



Figure 31. Confusion matrix (**a**) and ROC curve (**b**) of Classifier #5 with sigmoidal function kernel and dataset proportion of 80% for training and 20% for testing.

In relation to Classifier #5, for the produced AUC measures, it was observed that the classification with a proportion of 50% of the dataset used for training and testing presented the best result in relation to the classifiers set for this work placement, i.e., AUC = 45%, followed by the classification with a proportion of 80%:20% of the dataset used for training and testing, respectively, with AUC = 42%. The classification with 70%:30% of the dataset used for training testing, respectively, achieved a result of AUC = 34%.

It was also observed that, even in some cases where the metrics of the confusion matrix and the ROC curve showed considerable rates of false positives and false negatives, the classification rate of true-positive values was significantly more accurate. Such behavior can be explained by the fact that all images included the pest and that, even at different stages of development, its shape, size, and texture characteristics are similar.

Since the use of SVM classifiers was useful in the methodology for FAW recognition and classification, a deep analysis was carried out in order to select the best parameters. In terms of time consumption for training and testing, the best proportion among the presented results is 70% for training and 30% for testing, as illustrated in Table 12.

Additionally, analyses with the same proportions of data split were carried out on the CNN for FAW classification. In this scenario, for hidden layers, the ReLU function activation was selected and as the final activation function for the output layer, and the Softmax function was applied. The considered number of epochs was equal to twelve.

Tables 13, 14, and 15 present the results obtained using the A-CNN with dataset proportions of 50%:50%, 70%:30%, and 80%:20% for testing and testing, respectively.

SVM Classifier	Kernel Function	Parameter C	Parameter Delta (δ)
#1	Sigmoidal	10	1,0
# 2	Sigmoidal	100	1,0
# 3	Sigmoidal	0.1	0.1
#4	Sigmoidal	0.1	10
# 5	Sigmoidal	1	1

Table 12. Selected SVM classifiers for fall armyworm (Spodoptera frugperda) image pattern classification.

Table 13. Fall armyworm stage classification with A-CNN and dataset proportion of 50% for training and testing.

Instar	Precision	Recall	F1 Score	Accuracy (%)
#1	0.84	0.83	0.83	0.90
# 2	0.96	0.69	0.80	0.90
#3	0.80	1.00	0.89	0.90
#4	0.95	0.95	0.95	0.90
# 5	1.00	1.00	1.00	0.90

Table 14. Fall armyworm stage classification with A-CNN and dataset proportion of 70% for training and 30% for testing.

Instar	Precision	Recall	F1 Score	Accuracy (%)
#1	0.55	0.97	0.70	0.83
# 2	1.00	0.13	0.23	0.83
#3	1.00	0.99	0.99	0.83
#4	0.94	1.00	0.97	0.83
# 5	1.00	1.00	1.00	0.83

Instar	Precision	Recall	F1 Score	Accuracy (%)
# 1	0.88	0.94	0.91	0.84
# 2	1.00	0.80	0.89	0.84
#3	0.60	1.00	0.85	0.84
#4	0.98	0.51	0.67	0.84
# 5	1.00	0.97	0.98	0.84

Table 15. Fall armyworm stage classification with A-CNN and dataset proportions of 80% for training and 20% for testing.

Figures 32–34 illustrate the confusion matrices and ROC curves for the results obtained using the A-CNN.

Tables 16, 17, and 18 present comparisons of SVM classifiers and the A-CNN based on precision and accuracy, with dataset proportions of 50%:50%, 70%:30%, and 80%:20% training and testing, respectively.

Table 16. Comparison of fall armyworm classification with SVM and A-CNN using a dataset proportion of 50% for training and testing.

Instar	SVM Accuracy (%)	CNN Accuracy (%)	SMV Precision	CNN Precision
# 1	0.72	0.90	0.80	0.84
# 2	0.68	0.90	0.80	0.96
#3	0.80	0.90	0.80	0.80
#4	0.80	0.90	0.91	0.95
# 5	0.71	0.90	0.80	1.00

Table 17. Comparison of fall armyworm classification with SVM and the A-CNN using dataset proportions of 70% for training and 30% for testing.

Instar	SVM Accuracy (%)	CNN Accuracy (%)	SMV Precision	CNN Precision
#1	0.67	0.83	0.75	0.55
# 2	0.68	0.83	0.77	1.00
#3	0.77	0.83	0.77	1.00
#4	0.77	0.83	0.77	0.94
# 5	0.89	0.83	0.75	1.00

Table 18. Comparison of fall armyworm classification with SVM and the A-CNN using dataset proportions of 80% for training and 20% for testing.

Instar	SVM Accuracy (%)	CNN Accuracy (%)	SMV Precision	CNN Precision
#1	0.70	0.84	0.77	0.88
# 2	0.69	0.84	0.80	1.00
#3	0.78	0.84	0.78	0.60
#4	0.76	0.84	0.78	0.98
# 5	0.68	0.84	0.77	1.00

For the configurations presented as options for the computational intelligence stage, both for the use of SVM classifiers (with an ML focus) and for the use of the A-CNN (with a DL focus), percentages of 50:50%, 70:30%, and 80:20% were considered for training and testing, respectively. For such a context, both the confusion matrix and the respective ROC



curves were observed in order to evaluate the information regarding precision and accuracy for all different instars of the FAW caterpillar.

Figure 32. Confusion matrix (**a**) and ROC curve (**b**) of the A-CNN classifier with dataset proportions of 50% for training and 50% for testing.



Figure 33. Confusion matrix (**a**) and ROC curve (**b**) of the A-CNN classifier with dataset proportions of 70% for training and 30% for testing.



Figure 34. Confusion matrix (**a**) and ROC curve (**b**) of the A-CNN classifier with dataset proportions of 80% for training and 20% for testing.

Taking into account these results, it was possible to observe the following. For instar #1 the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to an accuracy equal to 90% and a precision equal to 84%. For instar #2, the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to an accuracy equal to 90% and a precision equal to 96%. For instar #3, the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to an accuracy equal to 90% and a precision equal to 96%. For instar #3, the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to

an accuracy equal to 90% and a precision equal to 80%. It is important to observe that for instar #3, the resulting precision value, when using the SVM classifier, was equal to that achieved by the A-CNN; however, the accuracy value was smaller. For instar #4, the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to an accuracy equal to 90% and a precision equal to 95%. For instar #5, the best configuration was obtained using the A-CNN with a proportion of 50%:50% for testing and training, respectively, leading to an accuracy equal to 90% and a precision equal to 90% and a precision equal to 90% and a precision equal to 100%. Table 19 presents the final parametrization for the A-CNN to classify the different instars of FAW caterpillars.

Instar	Hidden-Layer Activation Function	Output-Layer Activation Function	Epochs
#1	ReLU	Softmax	10
# 2	ReLU	Softmax	10
#3	ReLU	Softmax	10
#4	ReLU	Softmax	10
# 5	ReLU	Softmax	10

Table 19. A-CNN parametrization for fall armyworm (Spodoptera frugperda) image pattern classification.

Furthermore, Figure 35 illustrates the resultant context analysis considering the use of both ML and DL for FAW caterpillar classification purposes, focusing its control in a maize crop area. It was possible to observe that the structure that considers ML with SVM classifiers solved the problem in a good way, including gains in performance; however, A-CNN showed much better results.



Figure 35. Comparative time-consumption analysis between ML and CNN training and testing for FAW stage identification.

The use of DL has been increasing in recent years, allowing for a multiplicity of data analyses from different angles. Therefore, DL algorithms are recommended problems that require multiple solutions or that may depend more heavily on situations that require the leveraging of technologies to solve problems that involve decisions based on unstructured or unlabeled data.

Although the use of ML based on structured data enabled a solution, including facilities for interoperability, the use of one A-CNN allows for a robust decision support system for FAW caterpillar classification. Additionally, such a result can be coupled with

an agricultural fungicide sprayer to control varying dose rates as a function of FAW instars, i.e., enabling pest control in maize plants.

Another relevant aspect observed in this contextual analysis is that the use of ML required less training time compared to the same percentage of samples used in DL, which could be of interest for the scope of the problem related to pattern recognition and dynamic classification of FAW caterpillars in maize crops, leading to the opportunity to use less expensive hardware. However, today, one may use advanced hardware, such as Field Programmable Gate Arrays (FPGAs) or even Graphical Processor Units (GPUs) for time acceleration, which can bring about significant reductions in time processing, i.e., making the use of DL necessary. In fact, the experiments conducted for validation of the method proposed herein show its capacity to classify the patterns presented by FAWs in maize crops, which involves observing their different color, shape, size, and texture characteristics. The spatial location on maize plants should also be considered, i.e., whether present on the leaves or on the cobs.

4. Conclusions

An innovative method for in situ FAW recognition and classification is presented. The proposed method has the ability to evaluate the growth stages of this caterpillar species directly on maize plants. For validation, digital images of different stages of growth and different quantities of caterpillars were evaluated. Likewise, for image processing, techniques like filtering, segmentation by color scales, feature extraction based on the HOG and Hu algorithms, and the use of multivariate statistics with PCA were considered. Additionally, a contextual analysis for computational intelligence was conducted, taking into account not only an ML structure based on a set of SVM classifiers but also a DL supported by A-CNN. Results based on the evaluation of accuracy, precision, time processing, and hardware availability confirmed the DL structure with the A-CNN model as the final choice for the presented method, i.e., instead of the ML structure based on the use of SVM classifiers. The developed method has also proven to be useful in FAW pest control, which is considered the most dangerous pest for maize production, i.e., its use can allow for a decrease in losses, as well as the minimization of economic damage experienced farmers and producers. Im future work, one may consider the evolution of the developed method by including other computational intelligence techniques for pattern recognition and classification in both an unsupervised manner and through the use of a multi-spectral camera embedded in an Unmanned Aerial Vehicle (UAV) for real-time operation.

Author Contributions: This work was conducted collaboratively by both authors. Conceptualization, A.B.B. and P.E.C.; Formal analysis, A.B.B. and P.E.C.; Writing—original draft proposition A.B.B.; Writing—review and editing, P.E.C.; Supervision, P.E.C. All authors have read and agreed to the submitted version of the manuscript.

Funding: This work was supported by Embrapa Instrumentation and Fapesp (project number 17/19350-2).

Data Availability Statement: The original data presented in the study are openly available in repository 3453148_ABB_PEC on GitHub®, which can be accessed at https://github.com/alexbertolla/3453148_ABB_PEC, available to be accessed since 15 January 2025.

Acknowledgments: The authors thank the Brazilian Corporation for Agricultural Research (Embrapa) and the Post-Graduation Program in Computer Science of the Federal University of São Carlos (UFSCar).

Conflicts of Interest: The authors declare no conflicts of interests.

Abbreviations

The following abbreviations are used in this manuscript:

A-CNN	AlexNet Convolutional Neural Network
AI	Artificial Intelligence
BBS	Bacterial Brown Spot
BSR	Bacterial Soft Rot
CIE	Commission Internationale de l'Eclairage
CNN	Convolutional Neural Network
DL	Deep Learning
F-CNN	Faster Region-based Convolutional Neural Network
HOG	Histogram of Oriented Gradient
LDA	Linear Discriminant Analysis
MAP	Maximum A Posteriori
ML	Machine Learning
MSE	Mean Squared Error
NLM	Non-Local Mean
PBR	Phytophthora Black Rot
PCA	Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
R-FCN	Region-based Fully Convolutional Neural Network
RBF	Radial Basis Function
ROI	Region Of Interest
SNR	Signal-to-Noise Ratio
TN	Tensor Network
SVM	Support Vector Machine
USDA	United States Department of Agriculture

References

- 1. Kennett, D.J.; Prufer, K.M.; Culleton, B.J.; George, R.J.; Robinson, M.; Trask, W.R.; Buckley, G.M.; Moes, E.; Kate, E.J.; Harper, T.K.; et al. Early isotopic evidence for maize as a staple grain in the Americas. *Sci. Adv.* **2020**, *6*, eaba3245. [CrossRef] [PubMed]
- Erenstein, O.; Chamberlin, J.; Sonder, K. Estimating the global number and distribution of maize and wheat farms. *Glob. Food Sec.* 2021, *30*, 100558. [CrossRef]
- 3. USDA. World Agricultural Production; USDA: Washington, DC, USA, 2022.
- 4. Haque, M.A.; Marwaha, S.; Deb, C.K.; Nigam, S.; Arora, A.; Hooda, K.S.; Soujanya, P.L.; Aggarwal, S.K.; Lall, B.; Kumar, M.; et al. Deep learning-based approach for identification of diseases of maize crop. *Sci. Rep.* **2022**, *12*, 6334. [CrossRef]
- Viana, P.A.; Cruz, I.; Waquil, J.M. Árvore do Conhecimento do Milho: Pragas da Fase Inicial. 2011. Available online: https: //www.embrapa.br/en/agencia-de-informacao-tecnologica/cultivos/milho/producao/pragas-e-doencas/pragas (accessed on 5 January 2025).
- Mutyambai, D.M.; Niassy, S.; Calatayud, P.A.; Subramanian, S. Agronomic factors influencing fall armyworm (*Spodoptera frugiperda*) infestation and damage and its co-occurrence with stemborers in maize cropping systems in Kenya. *Insects* 2022, 13, 266. [CrossRef] [PubMed]
- Makgoba, M.C.; Tshikhudo, P.P.; Nnzeru, L.R.; Makhado, R.A. Impact of fall armyworm (*Spodoptera frugiperda*)(JE Smith) on small-scale maize farmers and its control strategies in the Limpopo province, South Africa. *Jàmbá J. Disaster Risk Stud.* 2021, 13, 1016. [CrossRef]
- Horikoshi, R.J.; Vertuan, H.; de Castro, A.A.; Morrell, K.; Griffith, C.; Evans, A.; Tan, J.; Asiimwe, P.; Anderson, H.; José, M.O.; et al. A new generation of Bt maize for control of fall armyworm (*Spodoptera frugiperda*). *Pest Manag. Sci.* 2021, 77, 3727–3736. [CrossRef]
- 9. Divya, J.; Kalleshwaraswamy, C.; Mallikarjuna, H.; Deshmukh, S. Does recently invaded fall armyworm, *Spodoptera frugiperda* displace native lepidopteran pests of maize in India. *Curr. Sci.* **2021**, *120*, 1358–1367. [CrossRef]
- 10. Maino, J.L.; Schouten, R.; Overton, K.; Day, R.; Ekesi, S.; Bett, B.; Barton, M.; Gregg, P.C.; Umina, P.A.; Reynolds, O.L. Regional and seasonal activity predictions for fall armyworm in Australia. *Curr. Res. Insect Sci.* **2021**, *1*, 100010. [CrossRef]
- 11. European Food Safety Authority (EFSA); Kinkar, M.; Delbianco, A.; Vos, S. Pest survey card on *Spodoptera frugiperda*. *EFSA Support*. *Publ.* **2020**, *17*, 1895E.

- 12. Quick Guide-Fall Armyworm. Available online: https://www.mpi.govt.nz/dmsdocument/53053-Fall-Army-Work-Quick-Growers-Guide (accessed on 5 January 2025).
- Cruz-Esteban, S.; Valencia-Botín, A.J.; Virgen, A.; Santiesteban, A.; Mérida-Torres, N.M.; Rojas, J.C. Performance and efficiency of trap designs baited with sex pheromone for monitoring *Spodoptera frugiperda* males in corn crops. *Int. J. Trop. Insect Sci.* 2021, 42, 715–722. [CrossRef]
- Pfordt, A.; Paulus, S. A review on detection and differentiation of maize diseases and pests by imaging sensors. J. Plant Dis. Prot. 2025, 132, 40.
- 15. Sharma, N.; Sharma, R.; Jindal, N. Machine learning and deep learning applications-a vision. *Glob. Transitions Proc.* 2021, 2, 24–28.
- Shinde, P.P.; Shah, S. A review of machine learning and deep learning applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
- Valderrama Solis, M.A.; Valenzuela Nina, J.; Echaiz Espinoza, G.A.; Yanyachi Aco Cardenas, D.D.; Villanueva, J.M.M.; Salazar, A.O.; Villarreal, E.R.L. Innovative Machine Learning and Image Processing Methodology for Enhanced Detection of Aleurothrixus Floccosus. *Electronics* 2025, 14, 358. [CrossRef]
- 18. Duda, R.O.; Hart, P.E.; Stork, D.G. Pattern Classification; John Wiley & Sons: Hoboken, NJ, USA, 2012.
- Patel, D.J.; Bhatt, N. Insect identification among deep learning's meta-architectures using TensorFlow. Int. J. Eng. Adv. Technol. 2019, 9, 1910–1914.
- 20. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. Electron. Mark. 2021, 31, 685–695. [CrossRef]
- Xavier, A.d.C.; Sato, J.R.; Giraldi, G.A.; Rodrigues, P.S.; Thomaz, C.E. Classificação e extração de características discriminantes de imagens 2D de ultrassonografia mamária. In *Avanços em Visão Computacional*; Curitiba, P.R., Neves, L.A.P., Neto, H.V., Gonzaga, A., Eds.; Omnipax: Curitiba, Brazil, 2012; Chapter 4; pp. 65–87.
- Menke, A.B.; Junior, O.A.d.C.; Gomes, R.A.T.; Martins, É.d.S.; de Oliveira, S.N. Análise de mudanças do uso agrícola da terrra a partir de dados de sonsoriamento remoto multitemporal no município de Luis Eduardo Magalhães (BA-Brazil). *Rev. Bras. Ensino Fis.* 2017, *39*, 315–326. [CrossRef]
- 23. Sankaran, S.; Ehsani, R.; Etxeberria, E. Mid-infrared spectroscopy for detection of Huanglongbing (greening) in citrus leaves. *Talanta* **2010**, *83*, 574–581. [CrossRef]
- 24. Miranda, J.L.; Gerardo, B.D.; Tanguilig, B.T., III. Pest detection and extraction using Image processing techniques. *Int. J. Comput. Commun. Eng.* **2014**, *3*, 189–192. [CrossRef]
- Buades, A.; Coll, B.; Morel, J.M. A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, USA, 20–25 June 2005; Volume II, pp. 60–65. [CrossRef]
- 26. Mythili, C.; Kavitha, V. Efficient Technique for Color Image Noise Reduction. Res. Bull. Jordan ACM 2011, 2, 41–44.
- Mishra, R.; Mittal, N.; Khatri, S.K. Digital Image Restoration using Image Filtering Techniques. In Proceedings of the International Conference on Automation, Computational and Technology Management, London, UK, 24–26 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 268–272. [CrossRef]
- Bertolla, A.B.; Cruvinel, P.E. Band-Pass Filtering for Non-Stationary Noise in Agricultural Images to Pest Control Based on Adaptive Semantic Modeling. In Proceedings of the 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 27–29 January 2021; pp. 398–403. [CrossRef]
- He, Q.; Ma, B.; Qu, D.; Zhang, Q.; Hou, X.; Zhao, J. Cotton pests and diseases detection based on image processing. *TELKOMNIKA Indones. J. Electr. Eng.* 2013, 11, 3445–3450. [CrossRef]
- 30. Xia, C.; Chon, T.S.; Ren, Z.; Lee, J.M. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. *Ecol. Inform.* 2015, *29*, 139–146. [CrossRef]
- 31. Kumar, Y.; Dubey, A.K.; Jothi, A. Pest detection using adaptative thresholding. In Proceedings of the International Conference on Computing, Communication adn Automation (ICCCA2017), Greater Noida, India, 5–6 May 2017; pp. 42–46.
- Sriwastwa, A.; Prakash, S.; Mrinalini; Swarit, S.; Kumari, K.; Sahu, S.S. Detection of Pests Using Color Based Image Segmentation. In Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018, Coimbatore, India, 20–21 April 2018; pp. 1393–1396. [CrossRef]
- 33. Huang, K.Y. Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. *Comput. Electron. Agric.* 2007, 57, 3–11. [CrossRef]
- 34. Sette, P.G.C.; Maillard, P. Análise de textura de imagem de alta resolução para aprimorar a acurácia da classificação da mata atlântica no sul da Bahia. In Proceedings of the XV Simpósio Brasileiro de Sensoriamento Remoto, Curitiba, Brazil, 30 April–5 May 2011; p. 2020.
- 35. Elleithy, K. Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.

- Guerrero, J.M.; Pajares, G.; Montalvo, M.; Romeo, J.; Guijarro, M. Support Vector Machines for crop/weeds identification in maize fields. *Expert Syst. Appl.* 2012, 39, 11149–11155. [CrossRef]
- 37. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 2017, 17, 2022. [CrossRef] [PubMed]
- Lee, S.H.; Chan, C.S.; Wilkin, P.; Remagnino, P. Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the International Conference on Image Processing, ICIP, Quebec City, QC, Canada, 27–30 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 452–456. [CrossRef]
- Thenmozhi, K.; Reddy, U.S. Image processing techniques for insect shape detection in field crops. In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 699–704.
- 40. Tian, K.; Zeng, J.; Song, T.; Li, Z.; Evans, A.; Li, J. Tomato leaf diseases recognition based on deep convolutional neural networks. *J. Agric. Eng.* **2023**, *54*, 1432. [CrossRef]
- Evangelista, I.R.S. Bayesian wingbeat frequency classification and monitoring of flying insects using wireless sensor networks. In Proceedings of the IEEE Region 10 Annual International Conference, Proceedings/TENCON, Jeju, Republic of Korea, 28–31 October 2018; pp. 2403–2407. [CrossRef]
- 42. Nanda, M.A.; Seminar, K.B.; Nandika, D.; Maddu, A. A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information* **2018**, *9*, 5. [CrossRef]
- 43. Liu, X.; Zhang, R.; Meng, Z.; Hong, R.; Liu, G. On fusing the latent deep CNN feature for image classification. *World Wide Web* **2019**, 22, 423–436. [CrossRef]
- 44. Li, X.; Dai, B.; Sun, H.; Li, W. Corn classification system based on computer vision. Symmetry 2019, 11, 591. [CrossRef]
- 45. Abdelghafour, F.; Rosu, R.; Keresztes, B.; Germain, C.; da Costa, J.P. A Bayesian framework for joint structure and colour based pixel-wise classification of grapevine proximal images. *Comput. Electron. Agric.* **2019**, *158*, 345–357. [CrossRef]
- Moreno, B.M.; Cruvinel, P.E. Computer vision system for identifying on farming weed species. In Proceedings of the 2022 IEEE 16th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 26–28 January 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 287–292.
- 47. Wang, F.; Luo, Y. A Study on Corn Pest Detection Based on Improved YOLOv 7. In Proceedings of the 2024 7th International Conference on Computer Information Science and Application Technology (CISAT), Hangzhou, China, 12–14 July 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1039–1047.
- 48. Liu, J.; He, C.; Jiang, Y.; Wang, M.; Ye, Z.; He, M. A High-Precision Identification Method for Maize Leaf Diseases and Pests Based on LFMNet under Complex Backgrounds. *Plants* **2024**, *13*, 1827. [CrossRef]
- Zhong, M.; Li, Y.; Gao, Y. Research on Small-Target Detection of Flax Pests and Diseases in Natural Environment by Integrating Similarity-Aware Activation Module and Bidirectional Feature Pyramid Network Module Features. *Agronomy* 2025, 15, 187. [CrossRef]
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
- Abdurrazzaq, A.; Junoh, A.K.; Muhamad, W.Z.A.W.; Yahya, Z.; Mohd, I. An overview of multi-filters for eliminating impulse noise for digital images. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* 2020, 18, 385–393. [CrossRef]
- 52. Jain, P.; Tyagi, V. Spatial and Frequency Domain Filters for Restoration of Noisy Images. *IETE J. Educ.* 2013, 54, 108–116. [CrossRef]
- 53. Solomon, C.; Breckon, T. *Fundamentos de Processamento Digital de Imagens: Uma Abordagem Prática Com Exemplos em Matlab*; Grupo Gen-LTC: Rio de Janeiro, Brazil, 2000.
- 54. Said, A.B.; Hadjidj, R.; Eddine Melkemi, K.; Foufou, S. Multispectral image denoising with optimized vector non-local mean filter. *Digit. Signal Process. Rev. J.* 2016, *58*, 115–126. [CrossRef]
- 55. de Brito, A.R. Método para Classificação de Sementes Agrícolas em Imagens Obtidas por Tomografia de Raios-X em Alta Resolução. Ph.D. Thesis, Universidade Federal de São Carlos, São Carlos, Brazil, 2020.
- 56. Ibraheem, N.A.; Hasan, M.M.; Khan, R.Z.; Mishra, P.K. Understanding color models: A review. *ARPN J. Sci. Technol.* 2012, 2, 265–275.
- Saravanan, G.; Yamuna, G.; Nandhini, S. Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. In Proceedings of the International Conference on Communication and Signal Processing, ICCSP 2016, Melmaruvathur, India, 6–8 April 2016; pp. 462–466. [CrossRef]
- 58. Sangwine, S.J.; Horne, R.E. The Colour Image Processing Handbook; Chapman & Hall: London, UK, 1998.
- 59. Bansal, S.; Aggarwal, D. Color Image Segmentation using CIELab Color Space using Ant Colony Optimization. *Int. J. Comput. Appl.* **2011**, *29*, 28–34. [CrossRef]

- 60. Durmus, D. CIELAB Color space boundaries under theoretical spectra and 99 test color samples. *Color Res. Appl.* **2020**, *45*, 796–802. [CrossRef]
- 61. Kaur, A.; Kranthi, B.V. Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation. *Int. J. Appl. Inf. Syst.* **2012**, *3*, 30–33.
- 62. Baxes, G.A. Digital Image Processing: Principles and Applications; Wiley New York: New York, NY, USA, 1994.
- 63. Gonzalez, R.C.; Woods, R.C. Processamento Digital de Imagens; Pearson Educación: London, UK, 2009.
- 64. Kulkarni, N. Color thresholding method for image segmentation of natural images. *Int. J. Image Graph. Signal Process.* **2012**, *4*, 28–34. [CrossRef]
- 65. Otsu, N. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 1979, 9, 62-66. [CrossRef]
- 66. Gonzalez, R.C.; Woods, R.E. Processamento de Imagens Digitais, 3rd ed.; Editora Blucher: São Paulo, Brazil, 2010.
- 67. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume I, pp. 886–893. [CrossRef]
- 68. Hu, M.K. Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory 1962, 8, 179–187.
- 69. Chu, H.; Zhang, D.; Shao, Y.; Chang, Z.; Guo, Y.; Zhang, N. Using HOG Descriptors and UAV for Crop Pest Monitoring; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; Volume 1, pp. 1516–1519. [CrossRef]
- Zhao, W.; Wang, J. Study of feature extraction based visual invariance and species identification of weed seeds. In Proceedings of the 2010 Sixth International Conference on Natural Computation, Yantai, China, 10–12 August 2010; Volume 2, pp. 631–635. [CrossRef]
- Bertolla, A.B.; Cruvinel, P.E. Dimensionality Reduction for CCD Sensor-Based Image to Control Fall Armyworm in Agriculture. In Proceedings of the 2024 ALLSENSORS 9th International Conference on Advances in Sensors, Actuators, Metering and Sensing, Barcelona, Spain, 26–30 May 2024; IARIA: Bucharest, Romania, 2024; pp. 7–12.
- 72. Faceli, K.; Lorena, A.C.; Gama, J.; Carvalho, A.C.P.L.F. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*; LTC: Rio de Janeiro, Brazil, 2011.
- 73. Mitchell, T.M. Machine Learning; McGraw-Hill: New York, NY, USA, 1997; p. 432.
- 74. Haykin, S. Neural Networks: A Comprehensive Foundation; Prentice Hall PTR: Hoboken, NJ, USA, 1994.
- 75. Rimal, K.; Shah, K.; Jha, A. Advanced multi-class deep learning convolution neural network approach for insect pest classification using TensorFlow. *Int. J. Environ. Sci. Technol.* **2023**, *20*, 4003–4016.
- 76. Zekiwos, M.; Bruck, A. Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis. *J. Electr. Comput. Eng.* **2021**, 2021, 9981437.
- 77. Burges, C.J.C. A Tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1998; Volume 2, pp. 121–167.
- Campbell, C. An introduction to kernel methods. In *Studies in Fuzziness and Soft Computing*; PHYSICA-VERLAG: Heidelberg, Germany, 2001; Chapter 7; pp. 155–192.
- 79. Cortes, C.; Vapnik, V. Support-Vector Networks. Mach. Learn. 1995, 20, 273–297. [CrossRef]
- 80. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. IEEE Intell. Syst. Appl. 1998, 13, 18–28.
- 81. Lorena, A.C.; Carvalho, A.C.P.L.F.D. Uma introdução às support vector machines. RITA 2007, 14, 43-67.
- Ouamane, A.; Chouchane, A.; Himeur, Y.; Debilou, A.; Nadji, S.; Boubakeur, N.; Amira, A. Enhancing plant disease detection: A novel CNN-based approach with tensor subspace learning and HOWSVD-MDA. *Neural Comput. Appl.* 2024, 36, 22957–22981.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.