Sticky trap monitoring system using deep learning and machine learning algorithms

Muskan Raghav, Dr. Akhilesh Dubey*
* Department of Biological Sciences and Engineering, Netaji Subhas University of Technology, New Delhi- 110078, India.
E-mail (correspondence to) - akhilesh.dubey@nsut.ac.in

Abstract

India's economy heavily relies on agriculture, employing over 50% of the population and serving as the main source of income for many. The agricultural sector contributes approximately 16% of India's Gross Domestic Product (GDP) and is crucial for the rural economy. As concerns over the environmental impact of traditional pest management techniques and their associated costs increase, Integrated Pest Management (IPM) has become a popular and efficient alternative. Traditional pest identification methods are time-consuming and laborious, which highlights the importance of an automated insect detection and management system. This study aimed to develop a two-step system for automatically counting and identifying pests, overcoming the challenges posed by prior methods. The initial stage utilized the YOLOv5 object detection algorithm to identify and count insects on a trap, while the second stage involved classifying the detected insects into two species: Macrolophus pygmaeus and Trialeurodes vaporariorum (Whitefly). Four machine learning algorithms (Support Vector Machines, Naive Bayes Theorem, Decision Trees, and Random Forest) were utilized in the classification process to compare their performance, along with an unsupervised machine learning algorithm, K-means clustering.

Keywords *Integrated pest management, You Look Only Once (YOLO), Support Vector Machines, Naïve Bayes theorem, K means clustering, Decision trees, Random Forest*

1. Introduction

Agriculture plays a crucial role in driving the country's economic progress and shaping the overall standard of living (Kasinathan et al., 2020). However, severe losses in terms of yield and agricultural productivity are regularly observed in agricultural fields due to various factors. One of these factors includes pest attacks that hinder the development of agriculture by affecting the metabolic processes of crops (Zhong et al., 2018). Plant diseases and insect outbreaks are causing an average annual decline of 40% in global food supplies (Domingues et al., 2022). To combat these insect invasions, farmers pin their hopes on pesticides, but numerous studies have shown that pesticides are not a pragmatic and sustainable choice as they are detrimental to human health and the environment (Rani et al., 2020). Not only do they elevate health risks, but pesticides also have adverse impacts on the quality of soil, water, and habitat of wildlife (Baker et al., 2020).

Due to increasing concerns about the environmental impact of conventional pest control methods and the associated expenses, Integrated pest management (IPM) has emerged as a highly effective and precise approach to managing pest infestations. (Wen et al., 2012). An IPM approach is built upon the utilization of preventive measures for crop protection that relies on comprehensive knowledge of the environment, crop, pests, and natural enemies, as well as the application of efficient farming techniques to manage pests (Anderson et., 2019). Modern IPM involves not only the adoption of sensible use of pesticides through the concept of Economic Thresholds (ET) but also the integration of various pest management techniques, such as resistance crops, augmentative biological control (ABC), biotechnology, and others, in a harmonious manner (Bueno et al., 2021).

To effectively control pests, it's important to monitor their activity and population density. One simple method to do so in greenhouses is by using sticky paper traps. These traps have a sticky glue layer on a colored cardboard surface that attracts insects, allowing for quantitative information on their population and variety (Rustia et al., 2019). While sticky traps provide an affordable means of collecting insect samples, identifying the specimens manually is a laborious, time-consuming process and often requires entomologist experts (Cabrera et al., 2022). Conventional insect identification and classification techniques are tedious, fallible, and susceptible to errors (Kasinathan et al., 2021). Precision agriculture tools empower farmers to analyze the spatial-temporal variability of several key factors that impact plant health and productivity (Lima et al., 2021). Therefore, an automated insect identification system can benefit producers who have limited knowledge of pest scouting and own large farms (Wen et al., 2012).

Recent technological growth in imaging and computer vision have led to the development of image-based methods for detecting small-sized insects in controlled environment agriculture such as in greenhouses. These methods utilize both conventional and popular machine learning along with deep learning techniques (Wenyong et al., 2022). However, the majority of research that pertains to this subject has been conducted solely in a controlled laboratory setting. The application of this research analysis in real-world situations presents a more significant obstacle due to the differences in how images are obtained (Rustia et al., 2018).

Zhong et al., (2018) adopted a system described that uses a YOLO based deep learning method to detect and roughly calculate the number of insects, and employs an SVM to classify them. Furthermore, the system is designed to be implemented on a Raspberry PI. Solis-Sánchez et al. (2010) used shape features and an adaptive threshold discriminant method to detect whiteflies, while In their research, Xia et al. (2012) presented a multifractal analysis technique that involved using a mobile robot to capture insects and proposed a way to identify three frequently occurring pest species (aphid, thrips, and whitefly) in a greenhouse environment. Additionally, Xia et al. (2018) put forth an approach that utilized the watershed algorithm and Mahalanobis distance on the YCrCb color space, which resulted in a feasible and computationally efficient method for identifying pests. In their study, Gassoumi et al. (2007) utilized morphology features such as compactness, aspect ratio, and extent, to classify 12 common cotton pests. Other researchers have proposed various color-based segmentation and counting algorithms, but they were not robust to various conditions in the field, such as variable illumination and sticky glue degeneration.

On the other hand, deep learning algorithms have the ability to recognize and understand various aspects of data automatically by learning from the input data, which eliminates the need for complicated image processing methods and laborious feature engineering. Rustia et al. (2020) developed a sequential method that first identifies and removes non-insect objects from images and then categorizes the remaining insect objects into various species using a multi-class CNN classifier. Similarly, Li et al. (2021) introduced a deep learning technique based on the Faster R-CNN architecture to enhance the accuracy of detecting small insects in images captured by sticky traps. These techniques have great potential for improving pest detection and classification in greenhouse agriculture.

For successful recognition, it is necessary to extract suitable features. Typically, both global features (such as color, texture and morphology) and local features are chosen to assist in identification (Zhong et al., 2018). The main aim of this study is to propose a procedure for identifying small-sized insects using low-resolution images. The proposed method combines techniques for Machine learning and Deep learning, making it suitable for species identification. A detection and rough counting method based on the You Only Look Once (YOLO) object detection, as well as a classification method and counting using feature extraction based on Support Vector Machines (SVM) and other Machine learning algorithms, have been developed.

2. Material and Methods

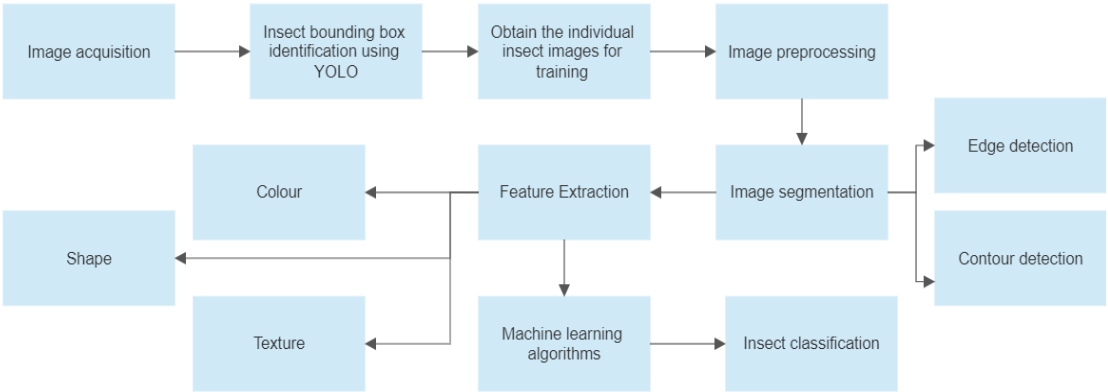

Figure 1 Flowchart of methodology

2.1 Image acquisition

The dataset used for training was obtained from the public repository on GitHub and was downloaded (Deserno et al., 2021). The dataset consisted of 284 JPEG images with a landscape orientation, measuring 5184 x 3456 pixels in size. The images depicted yellow sticky traps and were annotated with bounding boxes for three distinct categories of flying insects that are commonly found in greenhouse environments, namely *Macrolophus pygmaeus, Nesidiocoris tenuis,* and *Trialeurodes vaporariorum* (also referred to as Whitefly).

2.2 Insect identification using YOLO

The YOLO neural network is a convolutional model that uses a single-step approach to detect and classify objects. The YOLO algorithm works by taking an input image and passing it through a deep neural network that is composed of convolutional and fully connected layers. The network divides the input image into a grid of S X S cells and predicts bounding boxes for each cell that contain objects, along with the class probabilities of those objects. The YOLO algorithm uses a single neural network to predict all the objects within an image in one evaluation, hence the name "You Only Look Once." The YOLO algorithm uses anchor boxes to predict the bounding boxes for each object. These anchor boxes are pre-defined shapes that the algorithm uses to predict the size and shape of the object. During training, the algorithm adjusts the anchor boxes to improve the accuracy of the predictions. The YOLO algorithm also uses a concept called non-maximum suppression (NMS) to eliminate duplicate predictions. NMS is used to select the most confident bounding box when multiple boxes overlap the same object. The class-specific confidence score is calculated through network processing by multiplying the confidence score and the conditional class likelihood. In the end, YOLO employs the comparison of the confidence scores specific to each class to determine the results of object detection. (Kim et al., 2020).

Roboflow's cloud-based training infrastructure was used to train the obtained dataset. RoboFlow is an online tool that offers a dataset that's ideal for working with a small number of images. It allows users to create annotations by drawing a rectangle around objects in the image directly on the RoboFlow website, without the need for any additional downloads. RoboFlow offers a range of annotation formats and data augmentation functions, which can help to prevent overfitting during model training and save time during the training process (Park et al., 2022). The YOLOv5 version was selected to predict the bounding boxes around the insect. The purpose of this step was to detect the region of interest where the insect lies. A total of 284 images were uploaded on this platform and employed pre-trained COCO weights with 150 epochs. The overall dataset was sliced into a training dataset of 199 images, a set for validation of 57 images, and a set for testing of 28 images. After the training process was completed, various training graphs along with performance metrics were obtained.

For the purpose of comparison, YOLO object detection algorithm was also performed for insect classification.

2.3 Image acquisition and processing for Machine learning

In order to improve the performance of object classification, the SVM is provided with the bounding boxes of targets that were estimated by the YOLO model. A total of 1008 images of two different species, namely *Trialeurodes vaporariorum* (Whitefly), and *Macrolophus pygmaeus* (Table 1) were cropped from the whole sticky trap. The proposed approach utilized the Python programming and scripting language, along with the OpenCV image processing module, Google Colab cloud and GPU service, and the Jupyter Notebook. Image enhancement techniques, including sharpening, brightness, and contrast adjustments, were employed during the preprocessing stage to optimize the image quality as needed.

Table 1 The number of images of each species used for identification

Sno	Species	Number of images	Number of images after augmentation
1.	Trialeurodes vaporariorum (Whitefly)	300	600
2.	Macrolophus pygmaeus	204	408
	Total	604	1008

2.4 Image segmentation and processing

There are several segmentation techniques that are widely used, including thresholding, edge detection, region-based methods, and histogram-based methods. For instance, edge detection is utilized to locate the boundary between an insect image and its background by detecting the edges within the image (Thenmozhi et al., 2017). The dataset underwent preprocessing techniques, including resizing and augmentation such as flipping, rotation, cropping, brightness and saturation adjustments, and noise removal. 50 percent of the dataset was horizontally flipped, while rotation was conducted at angles ranging from ±45 degrees.

2.5 Feature extraction and Principal Component Analysis

The images were converted to binary, contours were identified, edges were detected using an algorithm and a mask was obtained. Feature extraction is implicated with mathematical tools for quantitatively defining an object. This work uses several features to create a feature space in order to gather a broad range of feature information. In this study, three sets of features were used to describe the insect's species: shape, color, and texture (Table 2). Numerous techniques exist for describing the texture of an image, with the Gray-level co-occurrence matrix (GLCM) features being among the most frequently employed texture features. The objective of this step was to convert the image into a list of numbers through which we can identify a particular insect. The 37 features were converted into major 2 components which are then fed into the machine learning algorithms.

Table 2 The features calculated

Sno	Feature category	Features
1	Shape	Area, Minor axis, Extent, Major axis, Perimeter, Eccentricity, Form Factor, Solidity, Aspect ratio, Average color
2	Color	Average color
3	Texture	Contrast, Dissimilarity, Homogeneity, Energy, Correlation, ASM

2.6 Machine learning algorithms

Based on the extracted features, we used various Machine Learning algorithms namely SVM, Naive Bayes Theorem, Decision tree, and Random forest to classify the detection results of YOLO into 2 classes. The classification performance of each algorithm was calculated using the Confusion matrix. Additionally, an unsupervised machine learning algorithm, K-means Clustering, was also performed.

3. Results and discussion

3.1 Insect identification and counting

A total of 284 sticky trap images were obtained from the online GitHub dataset. The images were divided into training, validation, and testing datasets. The YOLO v5 version object detector was adopted to detect and count the insects on a sticky trap using the Roboflow and Google Colab notebook. Various training graphs are shown in Figure 2. The results of identification where solid rectangles are used to mark the detected insect are shown in Figure

3. The performance metrics were as follows: the calculated precision was 83.8%, the mean average precision (mAP) was 86.8% and the recall was 84.5%. A total of 8,114 annotations were detected with an average of 28 insects per image. Average image size was 17.92 mp. This model was also be used to detect the number of insects and can provide a graph of changes in the number of insects w.r.t the number of days . Figure 4 demonstrates the output of the graph if four sticky trap images were uploaded in the code. Further, this model can be used to detect insects in an image that has never been encountered before. The images were then preprocessed to enhance the picture quality. As a result, the YOLO deep learning network demonstrates improved detection accuracy and better resistance to interference.

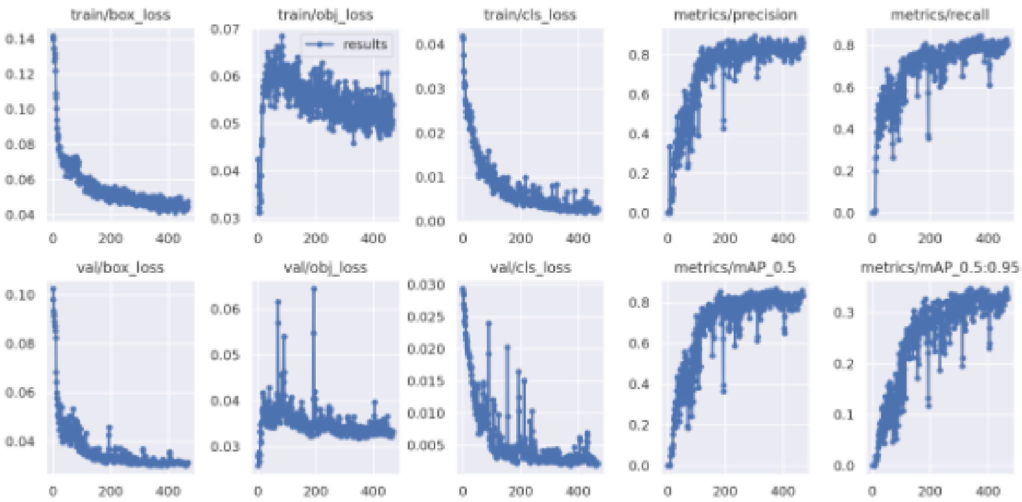

Figure 2 Training graphs

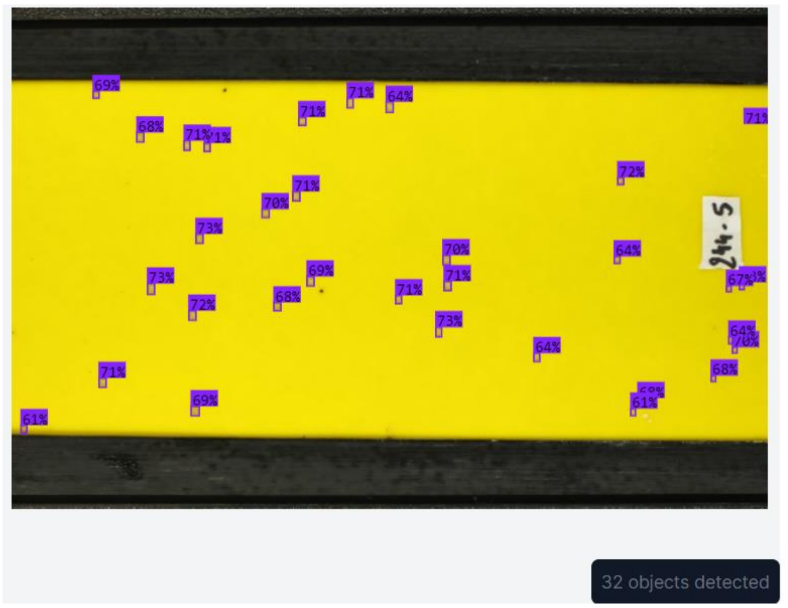

Figure 3 YOLO prediction results

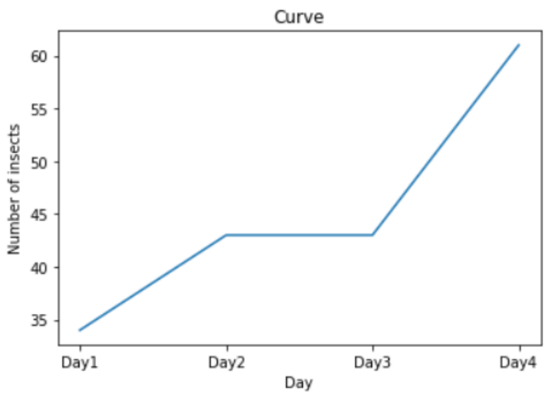

Figure 4 Graph showing an increase in the number of insects per day

3.2 YOLO insect classification

Roboflow was further also used for detecting the class of the object detected. Figure 5a shows the detection of *Macrolophus pygmaeus* detection and Figure 5b depicts the whitefly detection. The coordinates of the bounding box detected along with some other details such as confidence percentage and class are shown on the right side of the figure.

Figure 5 YOLO classification results (a) *Macrolophus pygmaeus* (b) *Trialeurodes vaporariorum* (Whitefly)

3.3 Image segmentation

Image segmentation is the process of transforming an image into multiple sections consisting of pixels that are identified by a label or a mask. Figure 6 depicts the results of various methods used for creating a mask of the image.

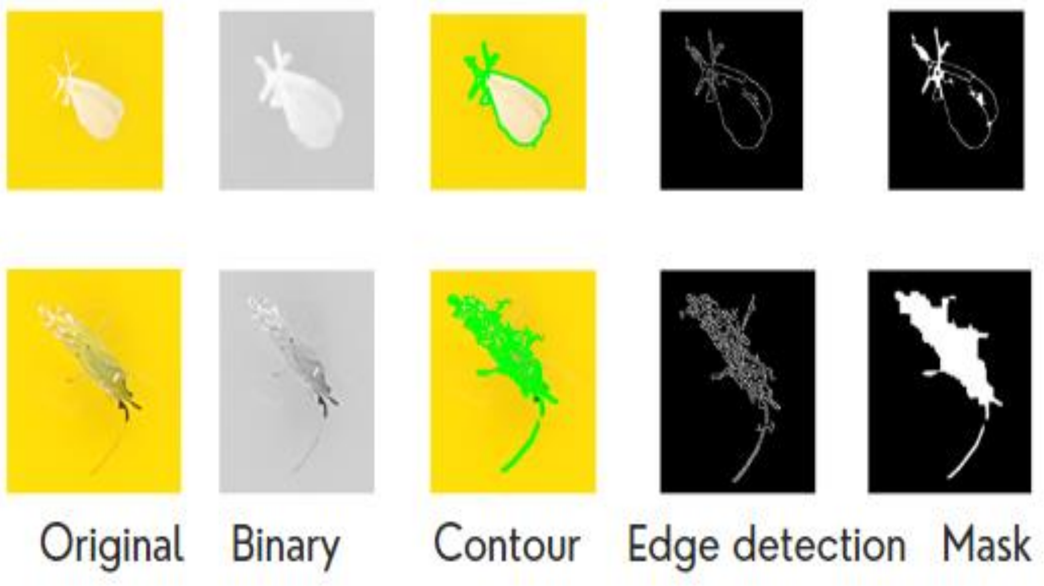

Figure 6 Image conversions

3.4 Feature extraction

For the purpose of feature extraction, the individual image of the insect was converted to binary, then edge detection using sobel techniques was done and finally, a mask of the image was obtained because of which the insect can be differentiated from the background. These conversions are shown in Figure 7. A total of 27 features were calculated and the values were stored in an excel sheet, a snippet of which is shown in Figure 5a and Figure 5b. As all these features are not possible to input in the Machine learning algorithms, Principal Component Analysis was done that converted the thirty-seven features to the most contributing two components. Now, these two components were fed to Machine learning algorithms. The complete list of features and values are available in Table S1 (Supplementary materials)

Shape features

Average colour (RGB)

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Name	Area	Perimeter	Aspect ratio	Extent	Solidity	Equi_dia	Form factor	Major axis	Minor axis	Eccentricity	Color_A	Color_B	Color__C
1.png	8667	376	0.75925926	0.979	1	105.048	0.7699859	107	81	1.3209877	11.447493	211.75723	251.78083
10.png	5751	304	1.13888889	0.974	1	85.571	0.7816023	71	81	0.8765432	13.862974	199.73103	250.34299
100.png	5851	306.8284	0.98717949	0.974	0.999829	86.3118	0.7806001	77	76	1.0131579	7.2187812	172.78422	218.56277
101.png	7600	352	1.31168831	0.977	1	98.3698	0.7704029	76	100	0.76	6.9170631	194.44182	234.01749
102.png	6075	312	0.92682927	0.975	1	87.9485	0.7838388	81	75	1.08	8.1639923	196.68646	236.96646
103.png	6956	336	0.78947368	0.976	1	94.1098	0.7738747	94	74	1.2702703	8.2509474	180.95635	206.0306
104.png	5320	292	0.92207792	0.973	1	82.3021	0.7836742	76	70	1.0857143	4.8388513	171.03219	203.21182
105.png	8544	370	0.92783505	0.979	1	104.3	0.7838761	96	89	1.0786517	12.019817	187.41879	210.44215
106.png	3732	251.6569	1.19298246	0.963	0.99467	68.9328	0.7401395	56	67	0.8358209	4.4726522	161.44014	205.31424
107.png	4620	278	1.51785714	0.971	1	76.6966	0.7508307	55	84	0.6547619	17.077521	195.65378	245.72038
108.png	4902	286	0.66666667	0.971	1	79.0027	0.7527155	86	57	1.5087719	15.507729	208.01427	246.5765
109.png	6399	320	1.025	0.975	1	90.2633	0.7848773	79	81	0.9753086	24.985518	203.46905	245.4189
11.png	7296	344	1.25974026	0.977	1	96.3823	0.7743862	76	96	0.7916667	12.496988	200.89731	249.5152
110.png	5700	302	1.01315789	0.974	1	85.1908	0.7849656	75	76	0.9868421	22.895933	202.47215	243.93404
111.png	5346	294	1.2238806	0.973	1	82.503	0.7768263	66	81	0.8148148	22.61103	198.49236	244.35202
112.png	5175	288	1.08571429	0.973	1	81.1727	0.7836372	69	75	0.92	22.357895	197.28665	241.57838
113.png	5396	294	0.93506494	0.973	1	82.8879	0.7840918	76	71	1.0704225	18.811508	195.35895	237.78626
114.png	6020	312	1.22535211	0.975	1	87.5494	0.7767423	70	86	0.8139535	22.797151	196.86353	241.41897
115.png	4386	274	0.59770115	0.969	1	74.729	0.7337653	86	51	1.6862745	21.530946	204.32714	243.65959
116.png	4554	270	1.04477612	0.971	1	76.1468	0.7846123	66	69	0.9565217	26.228785	198.29595	244.95864
117.png	4686	274	1.07462687	0.971	1	77.2425	0.7839544	66	71	0.9295775	22.065506	199.10075	244.20398
118.png	5580	304	1.44444444	0.973	1	84.2892	0.7583622	62	90	0.6888889	18.464678	190.57091	239.08756
119.png	5270	294	1.36507937	0.973	1	81.9144	0.7657828	62	85	0.7294118	7.7814692	170.84625	207.79199
12.png	9196	394	1.58441558	0.979	1	108.207	0.7440398	76	121	0.6280992	8.8553332	198.69395	247.74377

Contrast, Dissimilarity, Homogeneity, Energy, Correlation, ASM

O	P	Q	R	S	T	U	V	W	X	Y
Contrast_A	Contrast_B	Contrast_C	Contrast_D	Dissimilarity_	Dissimilarity_	Dissimilarity_	Dissimilarity_	Homogeneity_A	Homogeneity_B	Homogeneity_C
20.10597	18.24968	9.637338	31.56294	1.412780064	1.476404754	1.181673125	1.87008192	0.717857004	0.67290094	0.703421728
13.26492	37.99183	23.95929	25.21335	1.63117284	2.556598852	2.028340776	2.139106242	0.637796429	0.584676973	0.621051771
25.49916	35.33049	27.70973	51.65653	2.240721997	2.597402597	2.237982796	3.073479152	0.563724239	0.533605555	0.583181853
3.775455	5.593684	6.412064	12.71842	0.993376623	1.226842105	1.180693069	1.582368421	0.663637744	0.611579965	0.650799682
11.45886	13.43588	5.819201	17.34864	1.375284553	1.601481481	1.140838207	1.707160494	0.619587535	0.572606052	0.638501451
11.90697	15.62378	4.518582	13.0565	1.458036984	1.676682001	1.063829787	1.593300748	0.618243019	0.583355486	0.646171541
13.66345	15.07406	7.199778	21.83008	1.619294991	1.80075188	1.292809489	2.029699248	0.589404758	0.550295392	0.6091958
13.4367	14.88471	5.773727	18.94827	1.421406232	1.563085206	1.15775463	1.768960674	0.630202086	0.589574966	0.631183051
13.4907	30.12633	14.39338	16.97681	1.571353758	2.350746269	1.679621849	1.940565032	0.583383456	0.512985276	0.57521152
11.36373	22.29221	21.64257	31.02251	1.391794218	1.879220779	1.790588235	2.207359307	0.670016866	0.627365025	0.653200351
21.48941	30.4186	12.20369	25.98429	1.666465013	1.996328029	1.329190056	1.902284782	0.654201077	0.627331384	0.675392481
23.90216	17.63072	16.56098	43.63432	1.525	1.531020472	1.442111763	2.124394437	0.661375309	0.639612531	0.668612376
8.611337	26.10581	20.25258	24.36952	1.262310606	1.944901316	1.633749322	1.974232456	0.673209311	0.623168564	0.670051167
23.64318	17.51491	16.73835	44.34632	1.682306094	1.731052632	1.465281385	2.242807018	0.629156965	0.60256774	0.658899513
7.847061	27.58567	23.9126	27.65526	1.199926294	1.892068837	1.671286031	1.928357651	0.653937934	0.610256163	0.651722895
15.3139	47.19382	31.19794	27.52966	1.635047619	2.651207729	2.132341724	2.125990338	0.615149319	0.547406466	0.592554774
16.3715	26.46479	29.68202	48.44199	1.636180721	2.151964418	2.027046784	2.578762046	0.609228298	0.561212232	0.594469258
9.304782	17.49452	17.59475	26.91379	1.210776286	1.486212625	1.48407225	1.881893688	0.64753285	0.615849511	0.63045554
22.71963	36.13338	16.45438	24.94072	2.182330403	2.753989968	1.822898032	2.233926129	0.54498254	0.508237816	0.601800552
20.58447	20.61243	18.92403	39.84585	1.746917586	1.904040404	1.713636364	2.397013614	0.617087226	0.578119704	0.606347127
16.14652	18.45775	18.12942	35.14426	1.514189615	1.710840802	1.619739057	2.214255228	0.63334977	0.610244331	0.63038019
13.36543	35.82186	32.35271	41.53889	1.597883598	2.389605735	2.121588089	2.499103943	0.611451172	0.562893157	0.61233954
8.086835	26.29867	18.36328	19.73074	1.411951447	2.148766603	1.777006752	2.021062619	0.577318757	0.507253493	0.554230809
9.796179	27.66311	16.83456	16.72923	1.194697864	1.814267073	1.486626402	1.632231405	0.678308833	0.63202306	0.664338029
6.366808	19.57156	13.49761	14.52272	1.191331924	1.803398927	1.468965517	1.609302326	0.60569072	0.546987313	0.598193485
3.132123	7.865024	5.675357	7.421274	0.893281102	1.271995192	1.080595238	1.262379808	0.667756884	0.607211564	0.648716438
8.190631	18.61326	12.27243	17.48395	1.201714201	1.790579058	1.47563874	1.704170417	0.628600088	0.543597626	0.593054381

Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
Homogeneity_D	Energy_A	Energy_B	Energy_C	Energy_D	Correlation_A	Correlation_B	Correlation_C	Correlation_C	ASM_A	ASM_B	ASM_C	ASM_D
0.667304165	0.26699	0.24451	0.25535	0.24465	0.86272882	0.87634882	0.93407525	0.78614242	0.071	0.06	0.065	0.06
0.601001961	0.22096	0.20696	0.21851	0.20994	0.91934468	0.77159645	0.85458481	0.8484564	0.049	0.043	0.048	0.044
0.517934994	0.12437	0.11735	0.12695	0.11625	0.89222277	0.85176862	0.88296582	0.78325531	0.015	0.014	0.016	0.014
0.60308212	0.15015	0.13872	0.1498	0.13816	0.97253922	0.95960198	0.95340603	0.90814011	0.023	0.019	0.022	0.019
0.574397147	0.13123	0.12571	0.13755	0.12697	0.9318597	0.92067142	0.96556664	0.8974823	0.017	0.016	0.019	0.016
0.590829551	0.15484	0.14621	0.1582	0.14814	0.87819223	0.84176373	0.95377655	0.86775737	0.024	0.021	0.025	0.022
0.543690893	0.14711	0.138	0.14794	0.13705	0.93783297	0.93221142	0.96732633	0.90179608	0.022	0.019	0.022	0.019
0.588317228	0.14611	0.13594	0.14476	0.13676	0.91433839	0.90598556	0.96323569	0.88030075	0.021	0.018	0.021	0.019
0.526858671	0.11129	0.10088	0.11313	0.10442	0.92115205	0.82082361	0.91424241	0.89934707	0.012	0.01	0.013	0.011
0.610950038	0.21312	0.20254	0.21231	0.19864	0.88681855	0.78087845	0.78518128	0.69516351	0.045	0.041	0.045	0.039
0.627702638	0.191	0.18339	0.19665	0.18297	0.77287486	0.6819368	0.8706435	0.72832105	0.036	0.034	0.039	0.033
0.600521373	0.16417	0.15777	0.16626	0.14898	0.74284103	0.81161947	0.82190825	0.53380275	0.027	0.025	0.028	0.022
0.621231688	0.20745	0.19554	0.20638	0.19426	0.94919375	0.84765105	0.88107635	0.857785	0.043	0.038	0.043	0.038
0.583696099	0.15241	0.14536	0.15945	0.14315	0.81703162	0.86574444	0.87099907	0.66006473	0.023	0.021	0.025	0.02
0.596994121	0.15016	0.14159	0.15281	0.13799	0.91516981	0.70520725	0.74370646	0.70445451	0.023	0.02	0.023	0.019
0.57457878	0.15605	0.14309	0.15284	0.14686	0.87972506	0.63269515	0.75514514	0.78574172	0.024	0.02	0.023	0.022
0.535634257	0.13466	0.12511	0.13181	0.12211	0.91486989	0.86313597	0.8451523	0.74945924	0.018	0.016	0.017	0.015
0.568214528	0.15528	0.14882	0.15615	0.14286	0.92244837	0.85544903	0.85365832	0.77769486	0.024	0.022	0.024	0.02
0.54418915	0.15646	0.14907	0.16768	0.15404	0.88633928	0.82058363	0.91723065	0.87615731	0.024	0.022	0.028	0.024
0.559472679	0.15582	0.14725	0.15637	0.14557	0.79687514	0.79869351	0.81307154	0.61083195	0.024	0.022	0.024	0.021
0.556603623	0.14635	0.14045	0.14763	0.13357	0.8561452	0.83687783	0.83860191	0.68950888	0.021	0.02	0.022	0.018
0.563893854	0.12859	0.12111	0.13126	0.1209	0.92053016	0.78916385	0.80902379	0.75548733	0.017	0.015	0.017	0.015
0.517006292	0.11926	0.108	0.1156	0.10848	0.94568432	0.82526955	0.87752879	0.86892838	0.014	0.012	0.013	0.012
0.628340703	0.17593	0.16683	0.17481	0.1664	0.95015936	0.86052247	0.91472796	0.91565513	0.031	0.028	0.031	0.028
0.561244999	0.14437	0.13215	0.1428	0.13688	0.93270868	0.79565897	0.85822322	0.84835649	0.021	0.017	0.02	0.019
0.609408204	0.1468	0.13388	0.14287	0.13463	0.9646975	0.91224579	0.93636122	0.91717368	0.022	0.018	0.02	0.018
0.563668596	0.16074	0.14704	0.15731	0.14914	0.95462813	0.89769167	0.93218849	0.90389921	0.026	0.022	0.025	0.022

Figure 7 (a) shape and color features and (b) texture features

3.5 Machine learning algorithms evaluation metrics

For the Support Vector Machine and other algorithms, The red points, and the yellow area represent the species of *Macrolophus pygmaeus.* Similarly, the green points and the white area represent the species of whitefly. The accuracy came out to be 94.4%. Similarly, accuracy using Naive Bayes theorem, Decision Trees and Random Forest came out to be 94.04%, 94.4%, and 93.84% respectively. The results show that the Support Vector Machine algorithm showed the best results. Additionally using K means clustering, the two clusters of insects were obtained on a graph where both the species were plotted in blue and red points respectively. For the purpose of visualization, various graphs corresponding to all the algorithms are shown in Figure 9. The summary of results is shown in table 3.

The SVM is a non-parametric binary classifier that identifies the best hyperplane between two classes to distinguish them in new feature space with a high number of dimensions while considering solely the training samples that are located on the margins of the class distributions, which are referred to as support vectors (Moughal et al., 2013). When provided with an image that is represented by a collection of extracted features, and a set of predefined classes or groups of such features with known labels, NB employs a search method to identify the class to which the query features have the lowest total distance (Timofte et al., 2013).

The Decision Tree (DT) is one of the most widely recognized and oldest machine learning algorithms. It constructs a tree-like structure to model the decision-making logic, such as tests and corresponding outcomes, which is used to classify data items (Uddin et al., 2019). A Random Forest (RF) is a type of ensemble classifier that is composed of multiple decision trees, similar to how a forest is comprised of numerous trees (Uddin et al., 2019).

The key benefit of the YOLOv5 structure is that it enables objects to be located and classified in a single pass through the network. This feature allows for rapid frame-by-frame processing, making it feasible to perform real-time video analysis. To detect objects, the evaluation relied on three measures: precision, recall and mean average precision (mAP). Precision was calculated as the ratio of correctly identified objects to the total number of identified objects (error of commission), while recall was computed as the ratio of correctly identified objects to the total number of objects in the dataset (error of omission) (Sharma et al., 2022).(Figure 8)

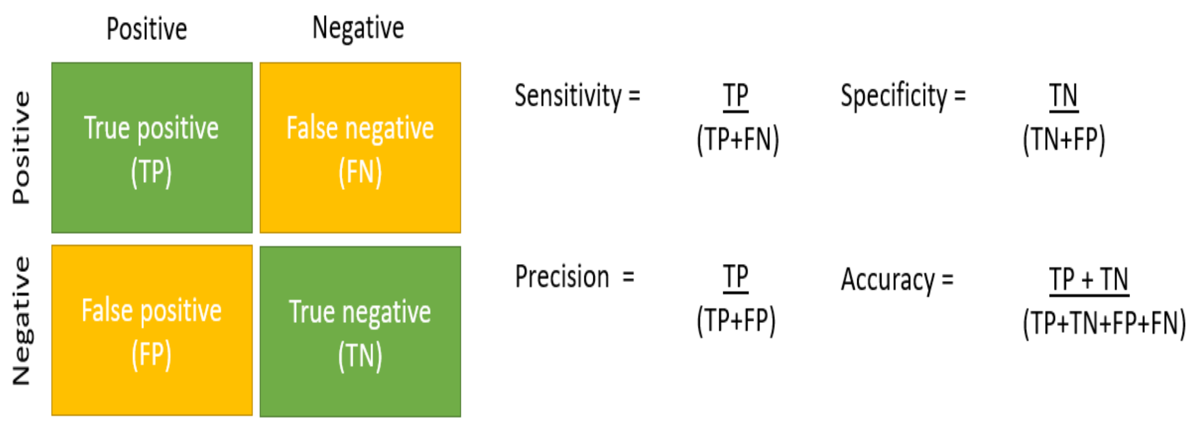

Figure 8 Evaluation metrics

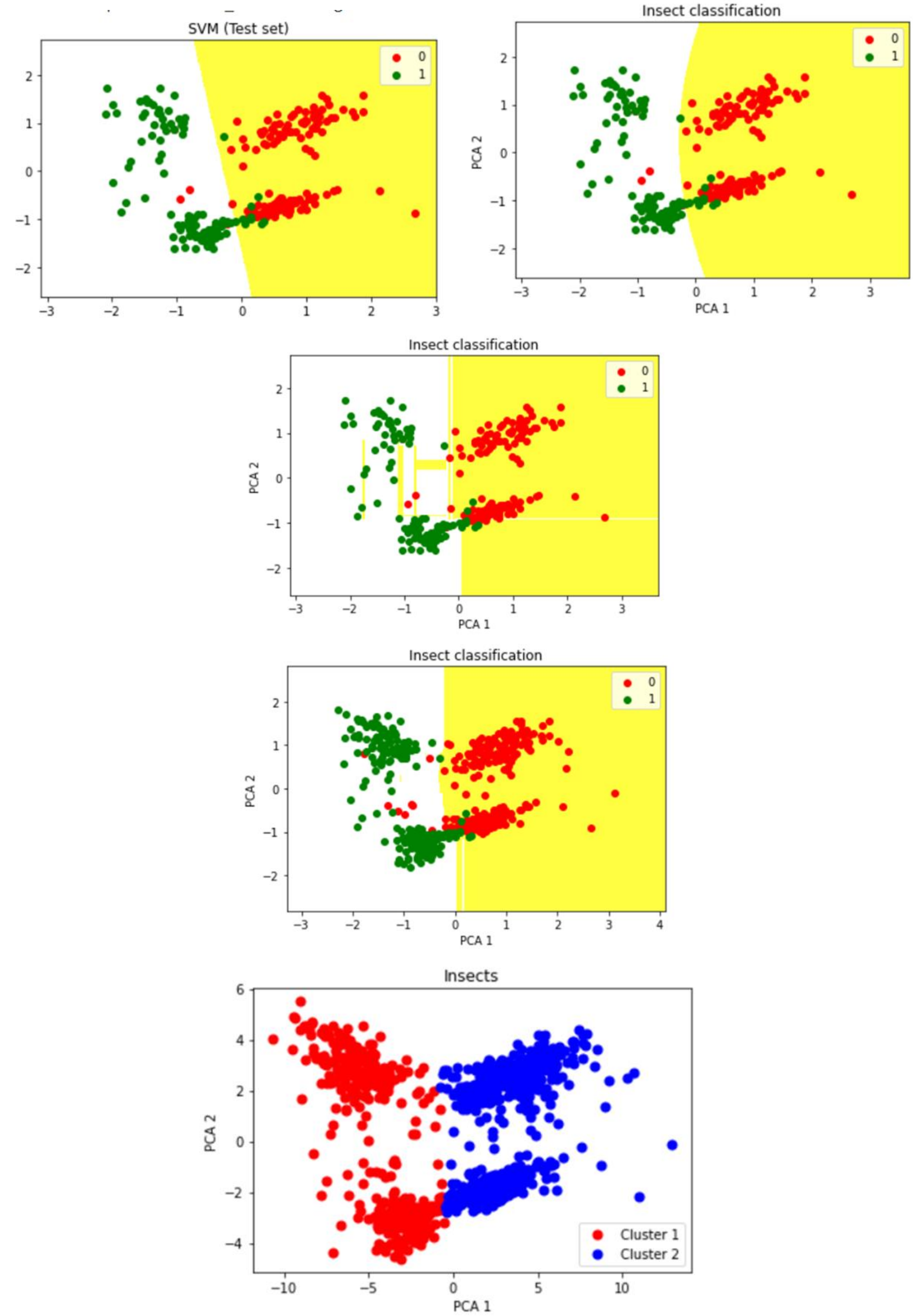

Figure 9 Visualization Plots of (a) Support Vector Machines (b) Naive Bayes theorem (c) Decision Trees, (d) Random Forest (e) K means clustering

Table 3 Metrics of all four Machine learning algorithms

Sno	Machine learning algorithm	Confusion matrix	Accuracy	Precision	Sensitivity	Specificity	Misclassification
1	Support Vector Machine (SVM)	[[145 4] [10 93]]	94.4%	97.31%	93.54%	95.87%	5.5%
2	Decision Tress	[[146 3] [11 92]]	94.4%	97.98%	92.99%	96.84%	5.5%
3	Random Forest	[[278 10] [21 195]]	93.84%	96.52%	92.97%	95.12%	6.1%
4	Naïve Bayes Theorem	[[145 4] [11 92]]	94.04%	97.31%	92.94%	95.83%	5.9%

4. Conclusion

This research presents a novel vision-based statistical and recognition system for insects that employs a YOLO deep learning network to detect and count insects, as well as an SVM and other Machine learning algorithms for classification. The system is designed to be implemented on a Google colab and Jupyter Notebook, and its effectiveness is verified using two different species of insects. The proposed system can detect and identify pests using a universal architecture, which can be easily modified to accommodate different insect categories. By combining the classification and fine-counting information of pests with meteorological and geographical data, the proposed system can provide real-time monitoring and forecasting of pest outbreaks, thereby enabling the prediction of suitable prevention and control measures for agricultural workers. This integrated service platform has significant potential for use in the field of agriculture.

References

1. Deserno, M. & Briassouli, A. (2021). Faster R-CNN and EfficientNet for Accurate Insect Identification in a Relabeled Yellow Sticky Traps Dataset. In 2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor) (pp. 209-214). doi: 10.1109/MetroAgriFor52389.2021.9628708
2. Kasinathan, Thenmozhi; Singaraju, Dakshayani; Uyyala, Srinivasulu Reddy (2020). Insect classification and detection in field crops using modern machine learning techniques. Information Processing in Agriculture, (), S2214317320302067–. doi:10.1016/j.inpa.2020.09.006
3. Kasinathan, T., Uyyala, S.R. Machine learning ensemble with image processing for pest identification and classification in field crops. Neural Comput & Applic 33, 7491–7504 (2021). https://doi.org/10.1007/s00521-020-05497-z
4. Wen, Chenglu; Guyer, Daniel (2012). Image-based orchard insect automated identification and classification method. Computers and Electronics in Agriculture, 89(), 110–115. doi:10.1016/j.compag.2012.08.008
5. Rustia, Dan Jeric Arcega; Erh Lin, Chien; Chung, Jui-Yung; Zhuang, Yi-Ji; Hsu, Ju-Chun; Lin, Ta-Te (2019). Application of an Image and Environmental Sensor Network for Automated Greenhouse Insect Pest Monitoring. Journal of Asia-Pacific Entomology, (), S1226861519304716–. doi:10.1016/j.aspen.2019.11.006
6. Cabrera, J., Villanueva, E. (2022). Investigating Generative Neural-Network Models for Building Pest Insect Detectors in Sticky Trap Images for the Peruvian Horticulture. In: , et al. Information Management and Big Data. SIMBig 2021. Communications in Computer and Information Science, vol 1577. Springer, Cham. https://doi.org/10.1007/978-3-031-04447-2_24
7. Rani, Lata; Thapa, Komal; Kanojia, Neha; Sharma, Neelam; Singh, Sukhbir; Grewal, Ajmer Singh; Srivastav, Arun Lal; Kaushal, Jyotsna (2020). An extensive review on the consequences of chemical pesticides on human health and environment. Journal of Cleaner Production, (), 124657–. doi:10.1016/j.jclepro.2020.124657
8. Domingues T, Brandão T, Ribeiro R, Ferreira JC. Insect Detection in Sticky Trap Images of Tomato Crops Using Machine Learning. Agriculture. 2022; 12(11):1967. https://doi.org/10.3390/agriculture12111967
9. Baker, Brian P.; Green, Thomas A.; Loker, Ali J. (2020). Biological control and integrated pest management in organic and conventional systems. Biological Control, 140(), 104095–. doi:10.1016/j.biocontrol.2019.104095
10. BUENO, A.F., PANIZZI, A.R., Hunt, T.E. et al. Challenges for Adoption of Integrated Pest Management (IPM): the Soybean Example. Neotrop Entomol 50, 5–20 (2021). https://doi.org/10.1007/s13744-020-00792-9
11. Anderson, J. A., Ellsworth, P. C., Faria, J. C., Head, G. P., Owen, M. D., Pilcher, C. D., ... & Meissle, M. (2019). Genetically engineered crops: importance of diversified integrated pest management for agricultural sustainability. Frontiers in bioengineering and biotechnology, 7, 24.

12. Li W, Yang Z, Lv J, Zheng T, Li M, Sun C. Detection of Small-Sized Insects in Sticky Trapping Images Using Spectral Residual Model and Machine Learning. Front Plant Sci. 2022 Jun 28;13:915543. doi: 10.3389/fpls.2022.915543. PMID: 35837447; PMCID: PMC9274131.
13. Zhong, Y., Gao, J., Lei, Q., & Zhou, Y. (2018). A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors, 18(5), 1489.
14. Bjerge, K., Alison, J., Dyrmann, M., Frigaard, C. E., Mann, H. M., & Høye, T. T. (2022). Accurate detection and identification of insects from camera trap images with deep learning. bioRxiv, 2022-10.
15. Wang, Pin; Fan, En; Wang, Peng (2020). Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning. Pattern Recognition Letters, (), S0167865520302981–. doi:10.1016/j.patrec.2020.07.042
16. Kim W, Cho H, Kim J, Kim B, Lee S. YOLO-Based Simultaneous Target Detection and Classification in Automotive FMCW Radar Systems. Sensors. 2020; 20(10):2897. https://doi.org/10.3390/s20102897
17. Sharma T, Debaque B, Duclos N, Chehri A, Kinder B, Fortier P. Deep Learning-Based Object Detection and Scene Perception under Bad Weather Conditions. Electronics. 2022; 11(4):563. https://doi.org/10.3390/electronics11040563
18. Timofte, R., Tuytelaars, T., Van Gool, L. (2013). Naive Bayes Image Classification: Beyond Nearest Neighbors. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds) Computer Vision – ACCV 2012. ACCV 2012. Lecture Notes in Computer Science, vol 7724. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37331-2_52
19. Moughal, T. A. (2013, June). Hyperspectral image classification using support vector machine. In journal of physics: conference series (Vol. 439, No. 1, p. 012042). IOP Publishing.
20. Uddin, S., Khan, A., Hossain, M. et al. Comparing different supervised machine learning algorithms for disease prediction. BMC Med Inform Decis Mak 19, 281 (2019). https://doi.org/10.1186/s12911-019-1004-8
21. Park, Y. H., & Jun, T. H. (2022). Classification of Soybean [Glycine max (L.) Merr.] Seed Based on Deep Learning Using the YOLOv5 Model. Plant breeding and biotechnology, 10(1), 75-80.
22. Thenmozhi, K., & Reddy, U. S. (2017, November). Image processing techniques for insect shape detection in field crops. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 699-704). IEEE.
23. Rustia, D. J. A., Lin, C. E., Chung, J. Y., & Lin, T. T. (2018, May). A real-time multi-class insect pest identification method using cascaded convolutional neural networks. In Proceedings of the 9th International Symposium on Machinery and Mechatronics for Agriculture and Biosystems Engineering (ISMAB), Jeju, Korea (Vol. 28).
24. Xia, Chunlei; Chon, Tae-Soo; Ren, Zongming; Lee, Jang-Myung (2014). Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. Ecological Informatics, (), S1574954114001228–. doi:10.1016/j.ecoinf.2014.09.006
25. Gassoumi, H.; Prasad, N.R.; Ellington, J.J. Neural Network-Based Approach for Insect Classification in Cotton Ecosystems; International Conference on Intelligent Technologies: Bangkok, Thailand, 2007.

26. Solis-Sánchez, L. O., García-Escalante, J. J., Castañeda-Miranda, R., Torres-Pacheco, I., and Guevara-González, R. (2010). Machine vision algorithm for whiteflies (Bemisia tabaci Genn.) scouting under greenhouse environment. J. Appl. Entomol. 133, 546–552.
27. Xia, C., Lee, J. M., Li, Y., Chung, B. K., & Chon, T. S. (2012). In situ detection of small-size insect pests sampled on traps using multifractal analysis. Optical Engineering, 51(2), 027001-027001.
28. Xia, D., Chen, P., Wang, B., Zhang, J., & Xie, C. (2018). Insect detection and classification based on an improved convolutional neural network. Sensors, 18(12), 4169.
29. Li, W., Wang, D., Li, M., Gao, Y., Wu, J., and Yang, X. (2021). Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. Comput. Electron. Agric. 183:106048. doi: 10.1016/j.compag.2021.106048
30. Rustia, D. J. A., Chao, J. J., Chiu, L. Y., Wu, Y. F., Chung, J. Y., Hsu, J. C., et al. (2020). Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. J. Appl. Entomol. 145, 1–17. doi: 10.1111/jen.12834