International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

Vol. 44 No. 1 April 2024, pp. 421-430

Spider Mite Detection: An Approach Of A Deep Convolutional Neuronal Network

Naram Isai Hernández Belmontes, Daniel Alaniz Lumbreras, Efrén González Ramírez, Hamurabi Gamboa Rosales.

Master in Science of Information Processing (MCPI)

Carretera Zacatecas-Guadalajara km 6, Ejido La Escondida.

C.P. 98160, Zacatecas, Zac.

Abstract— Spider mites pose a significant threat to tomato production worldwide, causing devastating yield losses. In Mexico, the world's second-largest tomato producer, these tiny pests are responsible for over 20% of crop loss. This challenge is further amplified by the presence of various tomato viruses. The pressure on the US tomato industry, facing fierce competition from imports, necessitates maintaining high-quality crops. Similarly, China's agricultural sector grapples with substantial damage caused by spider mites across numerous crops, including tomatoes. However, amidst this global concern, recent advancements in artificial intelligence (AI) offer a beacon of hope. Convolutional neural networks (CNNs), a cutting-edge deep learning technique, demonstrate remarkable promise in the early detection of spider mites on tomato plants. Research efforts utilizing CNNs are actively underway in countries like Mexico, India, Saudi Arabia, the USA, Turkey, and China. Our contribution to this global effort involved developing a streamlined CNN architecture specifically designed to enhance spider mite detection accuracy on tomato crops. This innovative approach achieved an impressive 97.56% accuracy rate, with a training time of just 13 minutes. This represents a significant reduction in training time compared to previous research. Furthermore, to ensure robust performance, it meticulously evaluated the model using additional metrics like sensitivity and AUC (Area Under the ROC Curve). This comprehensive approach underscores the effectiveness and reliability of our proposed CNN architecture. By highlighting the global impact of spider mites and the active research efforts across various countries, this revised paragraph emphasizes the urgency of the problem and positions your research as part of a larger international effort. Additionally, it delves deeper into the technical aspects of your CNN model, mentioning metrics like sensitivity and AUC, which demonstrates a more rigorous approach.

Keywords—Convolutional Neural Networks, Spider Mite, Deep Learning, Agriculture

I. Introduction

Agriculture, the very foundation upon which human civilization was built, has left an indelible mark on the course of our development. The momentous shift from nomadic lifestyles to settled societies, occurring roughly between 11,000 and 5,000 years Before the Present (BP), marked a pivotal moment. It was during this period that agricultural practices embarked on a continuous journey of evolution, forever intertwined with advancements in science and technology[1]. As agriculture matured and blossomed, various paradigms emerged, each one building upon the successes of the last. Today, it stands at the precipice of a new agricultural revolution – the era of precision agriculture, a data-driven management strategy poised to transform the way it cultivates crops. This pioneering approach hinges on the meticulous collection, thorough analysis, and seamless integration of real-time, spatially-specific, and individual data with a multitude of other relevant information sources. By

Corresponding Author: Naram Isai Hernández Belmontes

harnessing this rich tapestry of data, precision agriculture empowers farmers to make informed decisions that account for the inherent variability present within their fields[2]. Ultimately, this cutting-edge approach strives to achieve a multitude of objectives: optimizing the utilization of resources, significantly enhancing overall productivity, elevating crop quality to new heights, boosting profitability for farmers, and ensuring the long-term sustainability of agricultural production as a whole. Tomatoes, a crop of immense global significance, serve as a potent illustration of the immense potential that precision agriculture holds. Cultivated on a staggering expanse of nearly 4 million hectares (encompassing a staggering 3.989 million hectares), with a total yield exceeding a remarkable 108 million tons and an average yield surpassing an impressive 27,000 kg/ha[3], tomatoes thrive in diverse regions, including Mexico [4, 5], USA[6], China[7] and India[8]. Tomatoes are a testament to the power of human ingenuity in cultivation. However, despite the undeniable advancements witnessed in agricultural practices, a persistent challenge continues to plague farmers – the relentless battle against pests. Among these formidable adversaries, the red spider mite, scientifically classified as Tetranychus Urticae, stands out as a particularly pernicious foe, wreaking havoc on a variety of crops, including the tomato[9]. feeds primarily on tomato leaf undersides[9]. It even constructs intricate webs for protection[10]. This infestation translates to significant yield losses Traditionally, solutions included estimating economic damage thresholds[3], cultivating resistant tomato varieties[11], and applying synthetic acaricides[9] This paper proposes a method for identifying spider mites in tomatoes using a simple deep convolutional neural network (CNN). It compares it to existing methods to demonstrate that red mite detection can be achieved with a relatively basic architecture, potentially rivaling more complex models. The paper is meticulously structured to provide a comprehensive understanding of the proposed approach. Section 2 delves into related work on spider mite detection and management strategies, exploring the strengths and limitations of existing methods. Section 3 meticulously details the materials and methods employed in developing our deep learning-based approach, providing a step-by-step explanation of the model architecture, training process, and evaluation metrics. Section 4 presents the results obtained through experimentation, outlining the performance of our CNN model clearly and concisely. Section 5 follows with a comprehensive discussion of these findings, analyzing the results, exploring potential explanations, and identifying areas for future research. Finally, Section 6 summarizes our conclusions and outlines potential future directions for research in this domain, paving the way for further advancements in the field of precision agriculture through the continued integration of deep learning technologies.

II. Related work

The field of pest detection has witnessed a surge in research and development, particularly within the realm of computational sciences. Since the dawn of the 21st century, there has been a growing emphasis on leveraging these technologies for the classification of pests in various crops. This has led to a plethora of scientific investigations aimed at developing robust and accurate detection methods. One such study by Maeda et al. [12]. delves into the comparison of various convolutional neural network (CNN) architectures for classifying tomato pests. Their research explores the effectiveness of architectures like AlexNet, GoogleNet, InceptionV3, Residual Network 18 (ResNet-18), and ResNet-50. Notably, the evaluation utilized the PlantVillage database, a rich resource for training and testing image classification models in agriculture. The research meticulously assessed performance metrics such as accuracy, precision, sensitivity, specificity, F1-score, area under the curve (AUC), and receiver operating characteristic (ROC) curve. Interestingly, GoogleNet emerged as the frontrunner, achieving an impressive AUC of 99.72% and a sensitivity of 99.12%, demonstrating its exceptional ability to accurately detect tomato pests.

Instead, Agarwal et al. [11] presented a study focusing on CNNs specifically designed for pest detection in tomato leaves. Their research yielded promising results, with accuracy varying from 76% to 100% across different pest classes. Notably, the model proposed by another research group achieved an accuracy of 91.2%, highlighting the potential of this approach. Another study carried out by Chiluisa [13], explored the application of CNN algorithms for the detection of three prevalent tomato leaf pests: bacterial spot, early blight, and septoria leaf spot. Their work meticulously details the methodology employed to develop a robust classifier using CNN techniques.

Adding to this body of research, Krishnaswamy [14] conducted a study comparing the efficacy of two CNN architectures, VGG16 and AlexNet, for pest classification in tomato crops. Their investigation leveraged the PlantVillage database, encompassing images of healthy leaves, leaves infected with late blight, leaf mold, red spider mite, target spot, tomato mosaic virus, and tomato yellow leaf curl virus. The results showcased the remarkable accuracy of both CNNs, with VGG16 achieving 97.29% and AlexNet reaching 97.49% accuracy on a dataset of 13262 images. Notably, AlexNet secured the highest classification precision in this study. an algorithm of pest classification in tomato crops was performed, comparing in this case the use of two convolutional neural networks, the use of the VGG16 network and the AlexNet network, comparing the accuracy of each network. For the classification, the database provided by PlantVillage was where seven types of leaves of the plant are taken into account, which is, in a healthy state, with late blight, leaf mold, red spider, destination point, mosaic virus, and yellow curly virus. The results obtained show the accuracy of both convolutional neural networks, comparing the percentages of each neural network, the percentage of classification using 13262 images was 97.29% for VGG16 net and 97.49% for AlexNet being the latter the one that obtained the best precision. These advancements in AI-powered pest detection offer a promising future for precision agriculture. The ability to accurately identify and classify pests at an early stage empowers farmers to implement targeted interventions, minimize crop damage, and optimize the use of pesticides, leading to more sustainable and productive tomato cultivation.

TABLE I. MODEL COMPARASION.

Author	**Model**	**Accuracy**	**Sensitivity**	**Specificity**	**AUC**	**Time(min)**
Maeda et al.	Google Net	99.12%	99.93%	99.32%	99.72%	133.15
Argal et al.	Own Model	91.2%	N.A.	N.A.	N.A.	N.A.
Chiluisa	Own Model	93%	N.A.	N.A.	N.A.	N.A.
Krishnaswamy et al.	Own Model	97.49%	N.A.	N.A.	N.A.	N.A.

III. MATERIALS AND METHODS

This section delves into the exciting potential of artificial intelligence (AI), specifically deep learning techniques, for revolutionizing the early detection of spider mites in tomato crops. Our primary focus is on convolutional neural networks (CNNs). These powerful deep learning models excel at image recognition tasks, making them ideally suited for analyzing visual indicators of spider mite infestation. It will embark on a journey to understand the core mathematical principles that govern CNNs. Additionally, it will explore the practical implementation of these models using Python, empowering you to build your own AI-powered spider mite detection system. This exploration will equip farmers and agricultural professionals with a powerful tool for proactive pest management, ultimately leading to healthier tomato crops and increased yields.

A. OBTAINING THE DATABASE

The training of the convolutional neural network was carried out using the PlantVillage[15] database, which contains 54,323 images of 14 crop types and 38 pest types. It analyzed only the state of tomato leaves, where it is healthy, and where they are infected with the plague of the spider mite. It created a new database with only the spider mite images and health in tomatoes. The first class has 3752 and the second class has 1631. The description can be observed in the table 2.

TABLE II. DESCRIPTION OF DATABASE.

Classes	**Number of Images**
Spider Mite	3752
Health	1631
Total	5383

In figure 1 it can be seen the spider mites leaf and the healthy leaf. The main difference is the yellow spots in the first picture and the regions where these spots are.

Fig. 1. Spider Mite leaf and Health Leaf

B. PREPROCESSING OF DATA

Before convolutional neural network training is performed a method called data augmentation. This technique is applied in the two databases, which are the training and testing: in each, the images are rotated, operations such as approaches, turns, and certain characteristics are modified, to obtain more data and improve the precision of the convolutional neural network. It describes all this process in table 3 and it can observe the effect in the figure 2.

TABLE III. DATA AUGMENTATION PARAMETERS.

Parameter	**Value**
Rescale	1./255
Width shift range	-100 to 100
Height shift range	-100 to 100
Rotation range	120
Brightness range	0.2 and 1.5
Zoom range	0.3 and 1.5
Shear range	50

By incorporating data augmentation, it equips the CNN with a broader range of visual representations of spider mites, ultimately leading to a more robust and precise detection model.

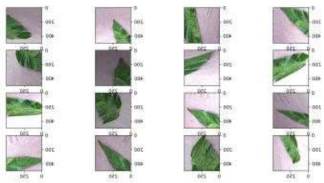

Fig. 2. Effects of data augmentation in leaf.

C. EXTRACTION OF CHARACTERISTICS

When using convolutional neural networks, in the first stage the convolutional layers are held, it is called convolution when it performs the multiplication of the nucleus with the pixels of the image, describes the convolution process in equation 1, where I is the image and K is the kernel[16]. This procedure performs the extraction of patterns and characteristics in the images entered. The first three convolutional layers are described below:

$$\boldsymbol{s(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n)} \qquad (1)$$

The first convolutional layer initially has 32 3x3x3 cores. The convolution is described in equation 1, the respective input images, the ReLu activation function is described in equation 2, and it’s displayed in figure 3. The last part of the first layer has the maximum grouping layer of 2x2.

$$f(x) = max(0, x) \qquad (2)$$

The second convolutional layer with 64 32x32 cores and similar to the previous layer uses the ReLu activation function, once finished, passes through the second maximum grouping layer similar to the previous 2x2, The third and last convolution layer, of 64 3x3 cores with the ReLu activation function [17] and the maximum grouping layer of 2x2, with this step you get the extraction of features to implement in the next part.

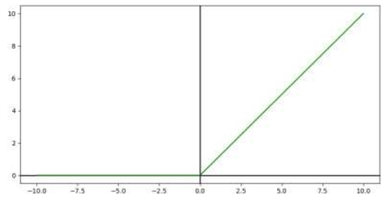

Fig. 3. ReLu Function.

D. ANALYSIS OF CLASSIFICATION

Once convolutions are performed for characteristic extraction a flattening layer is used to convert the matrix of numbers into a simple vector for use in the neural network. An optimizer called dropout is added, which consists of a regularization method for a wide variety of models[16], to deactivate some neurons randomly so that training weights are adjusted appropriately.

$$f(x) = \frac{1}{1 + e^{-x}} \qquad (3)$$

In this proposal a deactivation of 50 percent is used, then traditional neural networks are implemented with a layer of 256 neurons with the ReLu activation function, after the output layer with two classes and its sigmoid activation function described in equation 3 and observed in figure 4, the output values between 0 and 1 can be observed with this activation function.

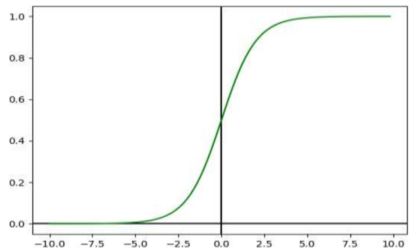

Fig. 4. Sigmoid Function.

E. TRAINING

Training in this CNN architecture is effective. All the experiments are carried out on a workstation, presenting the details in Table 5. The model used the Adam optimization algorithm. This was chosen because of is straightforward to implement, computationally efficient, has little memory requirements, and is invariant to diagonal rescaling of the gradients[18].

TABLE IV. ALGORITM PARAMETERS.

Hyper-Parameters	**Value**
Optimization algorithm	Adam
Epochs	90
Step per epochs	32

The training process was conducted by Python 3.6.0., with the library Tensorflow in version 2.4. 0., which provides libraries to design and implement CNN's, where applications and graphics help to visualize network activation and monitor the progress of network training. The loss function is binary cross entropy[19], is described in the equation 4.

$$J_{bce} = -\frac{1}{m}\sum_{m=1}^{M}[y_m \times \; log(h_\theta)(x_m) + 1(1 - \; y_m) \times log(1 - \; h_\theta(x_m))] \qquad (4)$$

Where

M number of training examples

y_m target label for training example m

x_m input for training example m

h_θ model with neural network weights θ

Also, the statistical analysis of this architecture was carried out with the package sklearn metrics and mxtend.

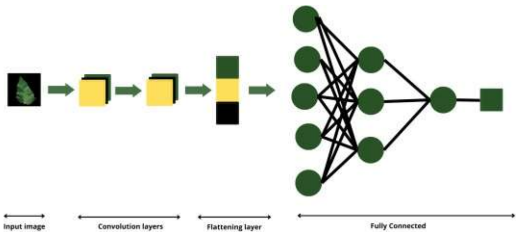

Fig. 5. Convolutional Neuronal Network Graph.

TABLE V. MACHINE SPECIFICATIONS

Hardware and Software	Characteristics
Memory	16 GB
Processor	Intel Core i7-7700HQ CPU @ 2.80 Ghz
Graphics	GeForce GTX 1060 X 8 Gb
Operating System	Windows 10, 64 bits

F. VALIDATION

d, which allows us to observe the truth positive (Vp), false positives (Fp), true negatives (Vn), and false negatives (Fn). This results in the metrics of sensitivity, specificity, accuracy, precision, and F1-Score [34], as well as the visualization of the model using the ROC curve. The first metric used for the evaluation of the convolutional neural network is sensitivity (recall) which consists in evaluating the ability of a model to predict the true positives, its formula is explained in the equation 5.

$$Sensitivity = \frac{Vp}{Vp + \; Fn} \qquad (5)$$

The second metric used is specificity, this evaluates the negativity of the results and is described in the equation 6.

$$Specificity = \frac{Vn}{Vn + Fp} \qquad (6)$$

The third metric used is accuracy, which evaluates how close the model is to true values, the formula is described in the equation 7.

$$Accuracy = \frac{Vp + Vn}{Vp + Fp + Vn + Fn} \qquad (7)$$

The fourth metric used is precision, which refers to the measurement of the values obtained and the degree of dispersion they possess, the less the accuracy, the greater this metric is obtained using equation 8.

$$Precision = \frac{Vp}{Vp + Fp} \qquad (8)$$

The fifth metric is the F1-Score, it is one of the most popular metrics because it throws accuracy and sensitivity into a single metric, its formula is described in the equation 9.

$$F1 - Score = \frac{Sensitivity * Precision}{Sensitivity + Precision} \qquad (9)$$

All of the metrics mentioned above are derived from the confounding matrix, as it is the common standard for the various models used in deep learning because they provide the information necessary to perform the corresponding analysis of the convolutional neural network. It can be seen in figure 6 obtained favorable results for the prediction of the detection of the red spider pest, obtaining 39 accurate predictions with the red spider, 40 predictions with the healthy leaf, plus get 0 errors in the healthy leaf and 1 in a red spider.

IV. RESULTS

The results obtained are favorable in the algorithm, an accuracy of 97.56%, a sensitivity of 99.6% was obtained, a specificity of 97.50%, and an F1-score of 98.77% with a 13-minute training time with 11 seconds and 23 milliseconds, reducing training time and retaining the percentage needed for model validation. In the figure 7 it can observer the regions that the algorithm detects, in some parts it can distingue the center part of the leaf, this is really important because, the spider mite presents mainly in this section, this happens for the sage leaf. In the figure 8, the accuracy increment in comparison of the loss, this happen with the number of the epochs. From epoch 60 to 80, it has a lot of perturbations in the training, but after this, the accuracy is maintained.

TABLE VI. METRICS OF THE EXPERIMENT

Author	Model	Accuracy	Sensitivity	Specificity	F1-Score	AUC	Time(min)
Hernandez et al.	Own Model	97.56%	99.6%	97.50%	98.77%	98.77%	12.05

As shown in table 6 the proposed model is the second with the highest percentage of precision, however, it is the one that works with the shortest time. This is because it is not a preloaded neural network that consumes a higher computational cost and that is reflected in the training time, therefore, it is possible to obtain a faster model that is in the metrics necessary for future implementation.

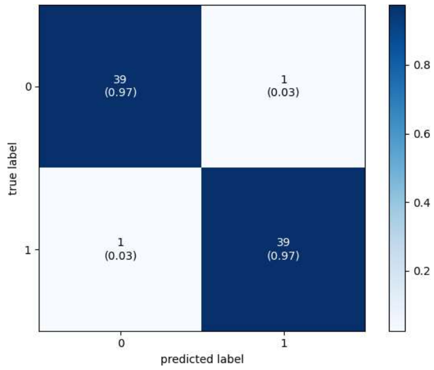

Fig. 6. Confusion Matrix.

V. DISCUSSION

Artificial intelligence is a powerful field in this kind of problem. Although the model is simple, the results obtained are excellent. Further testing is needed with various optimizers and configurations to get a good architecture. In the development of this research it was possible to create the classification model with a good percentage of accuracy. The use of convolutional neural networks is of vital importance for pest detection, since to identify the red spider, a series of recommendations should be followed to determine that the crop presents this pathology or is simply another disease or dehydration. For this with the use of Deep learning it is observed that the pattern in which the disease is presented with the database that was used for training, shows that the pest is concentrated in the leaf vein.

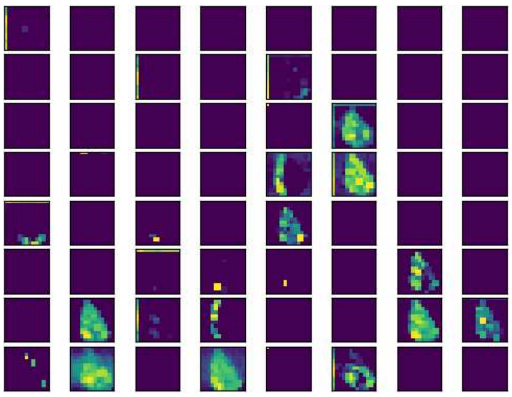

Fig. 7. Feature Extraction

VI. CONCLUSION

This research successfully developed a convolutional neural network (CNN) model with a high level of accuracy for classifying spider mite infestation in tomato leaves. CNNs are a powerful tool for pest detection, as traditional methods for identifying red spider mites often require complex procedures to distinguish them from other diseases or dehydration. Our deep learning approach leverages patterns within the training dataset, specifically focusing on the concentration of mites in leaf veins, to achieve accurate detection.

A significant advantage of our model is its reduced training time compared to previous studies. This was achieved by utilizing the ImageDataGenerator library for efficient image pre-processing. This library simplifies the process by handling tasks like rescaling, rotation, and zooming, eliminating the need for custom loop algorithms. Additionally, we employed a streamlined CNN architecture designed for faster training.

The model was implemented using Python and popular deep learning libraries like Keras, TensorFlow, and OpenCV. This combination of cutting-edge tools facilitated the development of a model capable of predicting spider mite infestation with an impressive 98% accuracy. Importantly, the model goes beyond simply classifying infestation. It focuses on specific areas like leaf veins, damaged sections, and edges, mimicking the evaluation techniques used by plant pathologists. One area for further exploration is the detection of leaf roughness, which currently provides an indicative approach. Future research will focus on refining this aspect through additional analyses and tests.

REFERENCES

[1] P. C. G. Marín y D. Z. Villarreal, «El origen de la agricultura, la domesticación de plantas y el establecimiento de corredores biológico-culturales en Mesoamérica», *Rev. Geogr. Agríc.*, n.º 41, pp. 85-113, 2008.

[2] R. Rios, «La Agricultura de Precisión: Una necesidad actual», *Rev. Ing. Agric.*, vol. 1, p. 11, 2021.

[3] J. G G y B. Mallik, «Growth Stage Based Economic Injury Levels for Two Spotted Spider Mite, Tetranychus urticae Koch (Acari, Tetranychidae) on Tomato, Lycopersicon esculentum Mill», *Trop. Agric. Res.*, vol. 22, ene. 2011, doi: 10.4038/tar.v22i1.2670.

[4] M. A. Barron y F. Rello, «The impact of the tomato agroindustry on the rural poor in Mexico», *Agric. Econ.*, vol. 23, n.º 3, pp. 289-297, 2000, doi: 10.1111/j.1574-0862.2000.tb00280.x.

[5] L. E. Padilla Bernal, A. Rumayor-Rodriguez, O. Veyna, y E. Reyes-Rivasrigue, «Competitiveness of Zacatecas (Mexico) Protected Agriculture: The Fresh Tomato Industry», *Int. Food Agribus. Manag. Rev.*, vol. 13, ene. 2010.

[6] S. Li, F. Wu, Z. Guan, y T. Luo, «How trade affects the US produce industry: the case of fresh tomatoes», *Int. Food Agribus. Manag. Rev.*, vol. 25, n.º 1, 2021, Accedido: 1 de diciembre de 2022. [En línea]. Disponible en: https://ideas.repec.org/a/ags/ifaamr/316366.html

[7] H. Hong, J. Lin, y F. Huang, «Tomato Disease Detection and Classification by Deep Learning», en *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, jun. 2020, pp. 25-29. doi: 10.1109/ICBAIE49996.2020.00012.

[8] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, y S. Gupta, «ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network», *Procedia Comput. Sci.*, vol. 167, pp. 293-301, ene. 2020, doi: 10.1016/j.procs.2020.03.225.

[9] M. Jakubowska, R. Dobosz, D. Zawada, y J. Kowalska, «A Review of Crop Protection Methods against the Twospotted Spider Mite—Tetranychus urticae Koch (Acari: Tetranychidae)—With Special Reference to Alternative Methods», *Agriculture*, vol. 12, n.º 7, Art. n.º 7, jul. 2022, doi: 10.3390/agriculture12070898.

[10] K. Oku, S. Magalhaes, y M. Dicke, «The presence of webbing affects the oviposition rate of two-spotted spider mites, Tetranychus urticae (Acari: Tetranychidae)», *Exp. Appl. Acarol.*, vol. 49, pp. 167-72, mar. 2009, doi: 10.1007/s10493-009- 9252-4.

[11] «Resistance of strawberry genotypes to the two-spotted spider mite, Tetranychus urticae (Acari: Tetranychidae) | Persian Journal of Acarology», abr. 2022, Accedido: 29 de noviembre de 2022. [En línea]. Disponible en: https://www.biotaxa.org/pja/article/view/70867

[12] V. Maeda-Gutiérrez *et al.*, «Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases», *Appl. Sci.*, vol. 10, n.º 4, Art. n.º 4, ene. 2020, doi: 10.3390/app10041245.

[13] G. D. Chiluisa González, «Detección de enfermedades en plantas de tomate a través del análisis computacional de sus hojas», bachelorThesis, 2021. Accedido: 18 de octubre de 2022. [En línea]. Disponible en: http://repositorio.utn.edu.ec/handle/123456789/11319

[14] A. K. Rangarajan, R. Purushothaman, y A. Ramesh, «Tomato crop disease classification using pre-trained deep learning algorithm», *Procedia Comput. Sci.*, vol. 133, pp. 1040-1047, ene. 2018, doi: 10.1016/j.procs.2018.07.070.

[15] D. P. Hughes y M. Salathe, «An open access repository of images on plant health to enable the development of mobile disease diagnostics». arXiv, 11 de abril de 2016. doi: 10.48550/arXiv.1511.08060.

[16] I. Goodfeellow, Y. Bengio, y A. Courville, *Deep learning*. MIT press, 2016.

[17] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.

[18] D. P. Kingma y J. Ba, «Adam: A Method for Stochastic Optimization». arXiv, 29 de enero de 2017. doi: 10.48550/arXiv.1412.6980.

[19] Y. Ho y S. Wookey, «The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling», *IEEE Access*, vol. 8, pp. 4806-4813, 2020, doi: 10.1109/ACCESS.2019.2962617.