

A system for the detection of pests in crops

Oscar Alexander López-Gorozabel¹
Universidad Técnica de Manabí - Ecuador
oscar.lopez@utm.edu.ec

Ricardo Orlando Malla-Valdiviezo²
Universidad Técnica de Manabí - Ecuador
ricardo.malla@utm.edu.ec

Gabriel Eduardo Morejón-López³
Universidad Técnica de Manabí - Ecuador
gabriel.morejon@utm.edu.ec

Miguel Ángel León-Bravo⁴
Universidad Técnica de Manabí - Ecuador
miguel.leon@utm.edu.ec

doi.org/10.33386/593dp.2024.1.1898

V9-N1 (ene-feb) 2024, pp 128 - 137 | Recibido: 06 de septiembre del 2023 - Aceptado: 24 de octubre del 2023 (2 ronda rev.)

1 Ingeniero en Sistemas Informáticos y Licenciado en Trabajo Social por la Universidad Técnica de Manabí. Máster en Ingeniería de Software y Sistemas Informáticos por la UNIR. .

ORCID: <https://orcid.org/0000-0002-0640-9953>

2 Magister en Informática Empresarial UNIANDES, Ing. En Sistemas Informáticos Universidad Técnica de Manabí, Docente Universidad Técnica de Manabí, Ex - Coordinador Zonal de TIC MSP – Zona 4, Ex - Coordinador de Metas Institucionales y Asesor Educativo MINEDUC Zona 4, Sub Secretario de atención intergeneracional MIES..

ORCID: <https://orcid.org/0000-0003-0841-7495>

3 Experiencia en el área de desarrollo, telecomunicaciones, Inteligencia Artificial, publicaciones sobre sistemas de comunicaciones mediante Luz visible, herramientas de monitoreo, análisis de datos con modelos de inteligencia artificial. .

ORCID: <https://orcid.org/0000-0001-8902-4583>

4 Ing. en Sistemas Informáticos , Máster en Tecnología Educativa y competencias Digitales. Docente docente contratado de en la Universidad Técnica de Manabí, Facultad de Ciencias Informáticas.

ORCID: <https://orcid.org/0000-0002-3435-2560>

Descargar para Mendeley y Zotero

RESUMEN

La presente investigación plantea la construcción de una aplicación web destinada a la detección de plagas, en su primera fase se ha propuesto la detección de la plaga de mosca blanca, una de las más recurrentes en los cultivos de Manabí, afectando principalmente al cultivo de plantas como tomate, pimiento, col y cucurbitáceas como calabaza, pepino y hortalizas de hoja como lechuga o perejil. Este proyecto busca convertirse en un agente de monitorización de cultivos, actuando de forma automática y eficaz en la detección de plagas mediante el procesamiento de imágenes, para lo cual se desarrollaron diversos algoritmos soportados por la librería ImageAI, con los que fue posible crear, entrenar y probar un modelo de detección. En cuanto al funcionamiento de la aplicación web, el usuario podrá crear una cuenta y una vez logueado podrá acceder al módulo de captura, donde podrá tomar o subir una fotografía para su respectivo análisis. Esta investigación se basa en el método bibliográfico y analítico, además la información es de fuentes confiables, tales como: IEEE, Dialnet, ACM, Google Scholar, Repositorios Institucionales. Para el desarrollo de la aplicación web se utilizó el lenguaje de programación Python para el Backend y tecnologías como HTML, W3Css y JavaScript para el Frontend. Posteriormente, se utilizó MySQL para crear la base de datos. El marco de trabajo utilizado para el desarrollo de la aplicación fue Scrum, debido a la versatilidad de su metodología. Finalmente, como resultado de este proyecto, se obtiene la primera versión de software funcional, con aspiraciones de mejora en futuras versiones. **Palabras clave:** detección de plagas, procesamiento de imágenes, modelo de detección de plagas, python.

ABSTRACT

The present investigation proposes the construction of a web application destined to the detection of pests, in its first phase the detection of the whitefly plague has been proposed, one of the most recurrent in the Manabí crops, mainly affecting the cultivation of plants such as tomato, pepper, cabbage and cucurbits such as pumpkin, cucumber and leafy vegetables such as lettuce or parsley. This project seeks to become a crop monitoring agent, acting automatically and effectively in the detection of pests through image processing, for which various algorithms supported by the ImageAI library were developed, with which it was possible to create, train and test a detection model. Regarding the operation of the web application, the user will be able to create an account and once logged in will be able to access the capture module, where they will be able to take or upload a photo for their respective analysis.

This research is based on the bibliographic and analytical method, in addition the information is from reliable sources, such as: IEEE, Dialnet, ACM, Google Scholar, Institutional Repositories. For the development of the web application, the Python programming language was used for the Backend and technologies such as HTML, W3Css and JavaScript for the Frontend. Subsequently, MySQL was used to create the database.

The framework used for the development of the application was Scrum, due to the versatility of its methodology. Finally, as a result of this project, the first version of functional software is obtained, with aspirations to improve in future versions.

Keywords: pests' detection, image processing, pests' detection model, python.

Introduction

Agricultural activity has become the sustenance of man worldwide, despite being directly or indirectly related to it, it has played an essential role in human and economic development, therefore, over the years various techniques with the aim of promoting it and consecutively increasing the production of crops that are in demand at the local, national and international level.

Shruthi et al. (2019) establishes that the accelerated increase of the population in the metropolis becomes an advantage for the agricultural communities, due to the fact that the production is maximized, and the harvested products have more output. But in some cases, farmers have losses, and this is due to the appearance of pests, which interrupt the growth and maturation of the fruit of our lands. This is where the hand of man appears, through the disinfection of plants that sometimes affects the quality of the fruit, despite the fact that technology has also been incorporated in the agricultural field to combat these pests, it has not been possible obtain an improvement in crop yield, due to the appearance of different diseases in plants, caused by very small and difficult to detect organisms, such as: larvae, fungi and viruses.

For this reason, the development of a web application for the detection of pests in crops was proposed, where Deep learning techniques are applied, which achieve greater efficiency and speed in the detection process (recognition and precision in the processing of images) in relation to manual deletion. Currently, the techniques used to detect diseases in crops using deep learning are: Convolutional Neural Networks (CNNs) and Swarm Intelligence (SI), among others (Maniyath et al., 2018).

For this research, the objective was the development of a responsive web application for the detection of pests, in its first version the existence of the whitefly plague is identified, using ImageAI, a Python library that is supported by Deep Learning and Computer Techniques Vision.

Agriculture 4.0: A New Agro-Technological Field

Agriculture 4.0 it's a mix between agriculture and technologies, this movement seeks to implement the analysis of characteristics established by experts in large quantities of crops, therefore, knowledge of various analysis techniques is required, for example image processing that allows capture information on established crop characteristics, which can be color, size, texture, shape, among others (De Martí, 2018).

Information Technologies used in agriculture, allow to accelerate the productivity of crops, which is related to the quality of the product, minimization of resources used for crops, among others. In addition, when referring to the productivity of a crop, the following aspects must be taken into account: cultivation practice used, dose used, pest inspection, type of surface, variation of nutrients in the soil, availability of water for irrigation (Landau et al., 2014).

It And Agriculture

With the rise of ITs, agriculture has benefited from the production aspect. These tools have undoubtedly improved the process of crop conservation, but there are still problems, however, in the field of Computer Science, one of the branches that has had the most impact due to the advances presented appears, it is Artificial Intelligence. and within it, the deployment of machine and deep learning techniques that have been used for various jobs in areas such as: manufacturing, agriculture, medicine, economics, among others (Nuñez et al., 2015).

In agriculture, image processing through machine or deep learning techniques allows obtaining accurate information, as long as an efficient algorithm is carried out, however, certain resources and digital equipment are required, such as: cameras, sensors, drones, among others. These devices allow capturing and generating data to be later analyzed by algorithms for relevant purposes, these can be existence of pests or diseases that are related to crop productivity, irrigation, humidity control and in general allows decision making to reduce production costs (Sotomayor et al., 2014).

According to the intervention of IT in agriculture, we could consider certain equipment such as cameras, sensors, drones, mobile applications, web, among others. In this sense, these have become a support tool for the farmer, because they are used for the control and administration of crops, which allows generating information that will help intelligent decision-making in planting procedures, harvest, and production (Llamas & Sulé, 2021).

Systematization Of Agricultural Processes Through Computer Vision

Computer vision is a scientific discipline that emerges from Artificial Intelligence and allows the extraction of features from an image, the VPC work methodology starts from the interpretation of an image through an array of numbers, which will then allow the classifier to recognize patterns or objects (Sucar & G. Gomez, 2017).

Deep And Machine Learning for Identification Topics

Automatic Learning (Machine Learning) is a branch of Artificial Intelligence, used in the development of automatic techniques that are programmable by the hand of man, but that can then develop on their own. Machine Learning allows computers to identify large-scale data patterns and perform predictive analysis of these (Andrade, 2018).

Deep Learning is a set of Machine Learning algorithms, used to develop more complex abstraction techniques and without human supervision, machines learn by themselves with much more efficiency, Deep Learning is an improvement of Machine Learning, like as improvements in predictive analysis, greater speed in the processing of high volumes of data, among others (Andrade, 2018).

Neural Networks in Image Processing

Artificial Neural Networks (ANN) have become a computing scheme that resembles the structure of the human nervous system. A neural network is made up of multiple processors, making it an intelligent system that has an algorithm for adjusting weights or independent parameters, used to estimate performance requirements (Otálora et al., 2021).

On the other hand, convolutional neural networks (CNN) have been considered within Deep Learning, as networks specifically developed to identify characteristics of large volumes of images. Convolutional neural networks are inspired by artificial neural networks, only with the novelty that they incorporate non-linear operations for image processing (Pusiol, 2014).

Figure 1

Objects detected by bounding boxes in images.



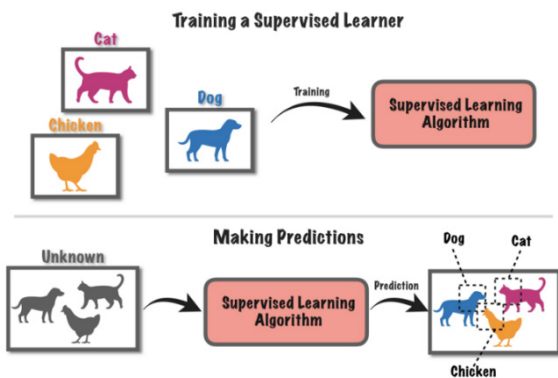
Logical Operation of Neural Networks

Neural networks are a set of neurons that are individually trained in a certain function and that generate a single output from N inputs. It should be noted that the most significant property of artificial neural networks (ANN) is their ability to learn from a group of training models, with which a data adjustment is established. The learning process of neural networks can occur through supervised learning and unsupervised learning (Sierra, 2022).

Supervised learning resides in the training of a network from a group of data made up of input and output patterns, where the learning algorithm must adjust the weights of the network so that the output is approximate to the input values of the network (Fuentes & Medina, 2021). Unsupervised learning presents only the set of input patterns to the artificial neural network, and the artificial neural network has the task of adjusting the weights of the network by implementing a given model. On the other hand, unsupervised learning allows the network to autonomously identify various characteristics, categories, etc., in input data and finally obtain the encoded data in the output (Rojas, 2020).

Figure 2

Neural network methodologies using supervised and unsupervised learning.



Python Image Processing

Image processing has become an elementary technique for the development of applications aimed at analyzing and obtaining information on the behavior of people, animals and objects with Python being the essential tool for its treatment due to the libraries it has to work with images and videos, these libraries allow the use of filters, gradients, effects, etc. Various modules and libraries can be used in Python for image processing, such as NumPy, SciPy, PIL, Scikit-image, SimpleCV, SimpleITK, Mahotas (Taquiá, 2017).

Object Detection in Python

Object detection in Python is part of the field of computer vision, which is considered to be a branch of artificial intelligence. Object detection is understood as the ability of the computer to identify characteristics of an image or video and from that to be able to make a value judgment. This ability is supported by computer programming techniques, in Python you can use a special library for object detection, which is OpenCV considered as a computer vision shelf and whose purpose is the manipulation of objects on the computer screen (Magaña, 2015).

Essential Libraries and Libraries for Image Processing or Object Detection with Python

Libraries in Python are containers of functional implements, which allow coding according to what the programmer requires, for example: implement specific functionalities, such as file input and output, graphics

visualization, work with machine learning and deep learning, processing natural language, numerical calculation, etc. All these activities require libraries that allow them to function for example, if you want to work with presentation of information in tables or data analysis, you can use the PANDAS library (Flores, 2021).

Tensorflow

TensorFlow is an open-source platform for machine learning or Machine Learning. This library has a comprehensive and adaptable environment of tools, libraries and resources that makes it easy for researchers to develop applications or functionalities with machine learning (Turned, 2017).

Keras

Keras is a library that allows working with neural networks, it can be executed on the TensorFlow platform, it was also developed with the aim of making experimentation with deep learning networks, which are part of Deep Learning, viable. Keras has extensive documentation and guides for beginners and developers who want to experiment with neural networks (Tanner, 2019).

Opencv

OpenCV is a computer vision library that enables image and video processing. It has more than 2,500 algorithms that can be accessed for free for commercial or academic purposes. OpenCV has a special version called opencv-python to merge with the Python programming language (Marín, 2022).

Imageai

ImageAI is an open-source library built in Python. This library allows developers to create applications and systems with autonomous deep learning and computer vision capabilities using a few lines of code (Image AI, 2022).

Yolov3

YOLOv3, is a family of object detection architectures and models pre-trained on the COCO dataset, used especially in the development of web or mobile applications that allow object detection in real time. In addition, it has custom tags for object detection and to use it

is enough to write a few lines of code (Yolov3, 2022).

Materials and Methods

For this research, the experimental and applied methodology was used, because the practical development of the web application was carried out. In addition, another method used was the analytical one since it allowed the structuring and analysis of theory and practice. The bibliographic method was also incorporated, correspondingly through bibliographical research carried out in articles, theses and web pages that are related to the present work.

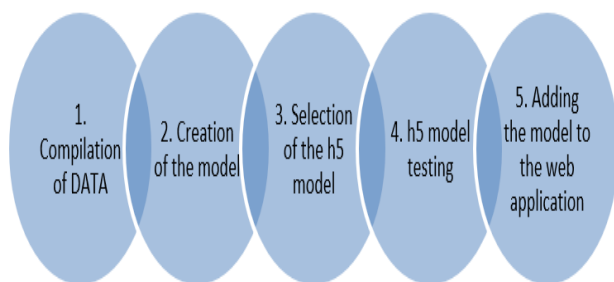
Results

Creating The Custom Detection Model with Python

Once the interfaces were built and validated, the Backend part of the application was developed using Python. Specifically, within the development, the Flask framework was used, which allowed migrating programming in Python to a web environment. For the pest detection part, an algorithm from the ImageAI library was implemented, which allowed the creation of a personalized model, to then train and test it. Within the development the following processes were carried out.

Figure 3

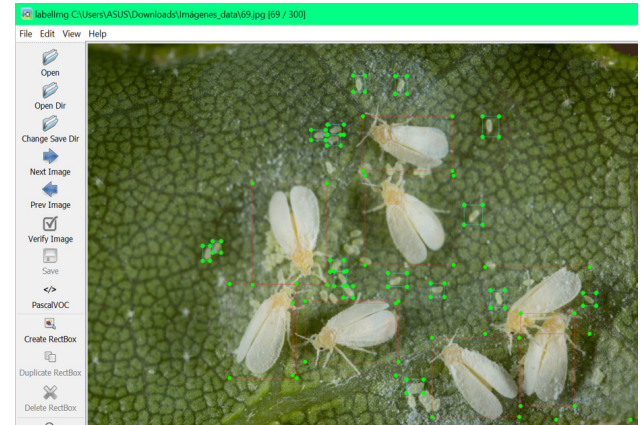
Phases for the creation of the h5 detection model



Labeling Images with The Labelimg Tool

Figure 4

Preparation of the images.

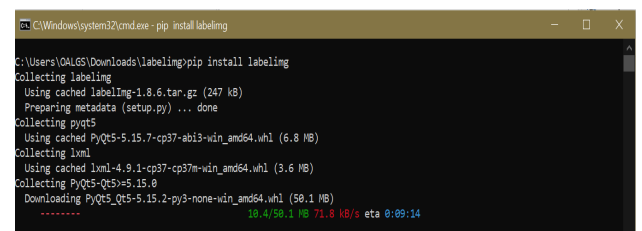


To start the detection, the insects (objects) within the images had to be previously labeled, in this first version the whitefly, for the labeling the free use tool labelimg was used, which allows the dimensions of the insects to be stored in XML files. that were captured in the images.

Preparation of the python environment, through the installation of libraries that allowed the execution of the system.

Figure 5

Installation of tools for labeling and processing.



In order to run the application without errors, it was necessary to install the required packages for the interfaces, file upload, image processing, training, camera activation, among others. These packages were:

- Tensorflow: used for managing neural networks.
- Keras: Tensorflow complement for image processing in neural networks.
- OpenCV: allows the use of the camera and the processing of images.

- Flask: allows the incorporation of HTML templates, typical of web development.
- Flask-MySQLdb: allows the integration of the application with the MySQL database manager.
- ImageAI: library that allows the detection of objects in images or video in real time.

Model Creation

Figure 6

Model programming.

```
''' PRIMER PASO '''
from imageai.Detection.Custom import DetectionModelTrainer
entrenamiento = DetectionModelTrainer()
entrenamiento.setModelTypeAsYOLOv3()
entrenamiento.setDataDirectory(data_directory="mosca")
entrenamiento.setTrainConfig(object_names_array=["mosca_blanca"],
batch_size=4,
num_experiments=25,
train_from_pretrained_model="pretrained-yolov3.h5")
entrenamiento.trainModel()
```

For the creation of a model, the classes are required in relation to the images that were labeled, then yolov3 is taken as a pre-training model. Finally, the Create_Model.py file was executed, where an optimal model for pest detection was obtained.

Model Selection

Figure 7

Code for selecting a detection model.

```
''' SEGUNDO PASO '''
from imageai.Detection.Custom import DetectionModelTrainer
entrenamiento = DetectionModelTrainer()
entrenamiento.setModelTypeAsYOLOv3()
entrenamiento.setDataDirectory(data_directory="mosca")
metrics = entrenamiento.evaluateModel(model_path="mosca/models",
json_path="mosca/json/detection_config.json",
iou_threshold=0.5, object_threshold=0.3,
nms_threshold=0.5)
print(metrics)
```

In order to select an accurate detection model, it is necessary to evaluate it based on the config.json file that stores the detection configuration after having trained the set of labeled images.

Model Testing

Figure 8

Test with a local image.

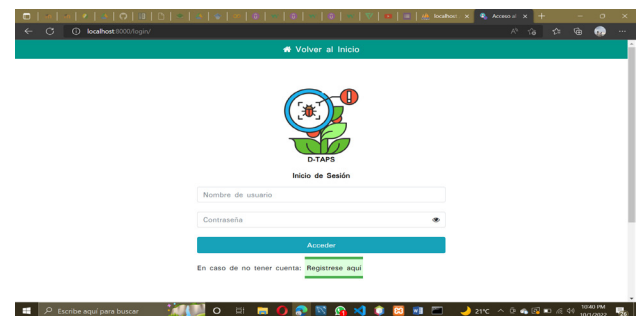
```
''' TERCER PASO '''
from imageai.Detection.Custom import CustomObjectDetection
detector = CustomObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath("detection_model-ex-021--loss-0023.644.h5")
detector.setJsonPath("detection_config.json")
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image="17.jpg", output_image_path="17_detected.jpg")
for detection in detections:
    print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])
```

For the detection in image or video, it is required that the inputs be established correctly, in the case of image, an input variable and an output variable must be created in order to generate a new image, which will be processed by the algorithm. An important piece of information is that the model generated in the training must be used for testing, which will always be found in the Models folder and will have the h5 extension. On the other hand, the .json file that contains the detection configuration must be placed.

Captures Of Interfaces of The Running Application

Figure 9

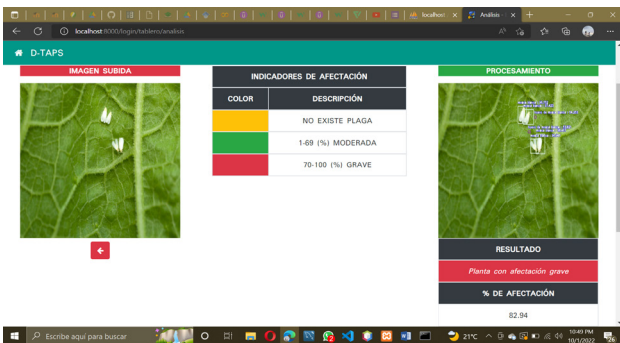
Login screen.



This interface allows access to the application by entering the username (email) and Password, both data are duly validated. In addition, the screen can adapt to any resolution, due to the adaptive design with which it was developed. On the other hand, when a user is not registered, they can proceed to do so from the Register here button.

Figure 10*Upload screen.*

In this section of the application, you can capture an image with the Mobile device's camera or also upload a file (photo) that has previously been stored in memory. Once it is selected, you can begin processing the image by pressing the Analyze button.

Figure 11*Results screen.*

After processing the image, the analysis interface is presented, where the percentage of affectation is shown, which allows it to be classified between one of the three affectation indicators (there is no pest when it is 0%, moderate when it is between 1-69% and serious between 70-100%). These values are generated from the execution of three Deep Learning algorithms that allowed the labeling, training, and validation of objects in images.

Conclusions

Currently, web applications have grown exponentially, and this is due to the emergence of various tools that allow efficient software development. Analyzing accuracy values in the context of web applications using Python and image processing and neural network tools is crucial to evaluate the performance and effectiveness of such applications. Accuracy refers to the measure of how close the predicted

values are to the actual values in a predictive model, such as a neural network model.

In this context, accuracy can be a critical factor, especially in applications that involve image recognition, data analysis and automated decision making. In this research, precision was taken as the evaluation metric, which can vary depending on the context and the data set used. This precision was calculated taking into account the number of true positives divided by the total number of positive predictions (true positives + false positives), these values must be close to 1.0 to be considered desirable, which indicates that the majority of affirmative detections are correct.

The results obtained in the various tests that were carried out were between 0.81 and 0.94, which allows us to determine that the algorithm has a high percentage of precision, thus ensuring that the detections have the lowest possible error. In addition, images of plants without pests were also processed and the result was 0.0, which is why it was labeled within the indicator (there is no pest).

The detection of images within the agricultural sector is especially important because it allows farmers to identify the characteristics of the crops more accurately, in order to improve the production of their crops and in turn avoid possible inconveniences with the pests that appear at each station. As a result, a web application was obtained that incorporates an algorithm for the precise detection of the whitefly plague and its tests were positive, since it was possible to detect it through the ImageAI library, thanks to its simple methodology that consisted of:

- Prepare the images (Labelling with labelImg).
- Use of Yolov3 pre-trained model.
- Creation and training of a new model based on Yolov3.
- Model tests using an image or video.
- Test the model using the device's camera in real time.

As future work is proposed:

- Incorporate other classes of pests for analysis.

- Real-time video capture.

Acknowledgment

We thank God and our families for their unconditional support.

Bibliographic references

Andrade, J. (2018). Aplicación móvil para la detección y tratamiento de daños de los cultivos de la parroquia Taura del cantón Durán, mediante el uso de software de análisis de imagen basado en técnicas de Machine Learning. <http://201.159.223.180/bitstream/3317/11769/1/T-UCSG-PRE-ART-IPM-150.pdf>

De Martí, S. P. (2018). AGRICULTURA DE PRECISIÓN y PROTECCIÓN DE CULTIVOS. *Revista de ingeniería*, 47, 10-19. <https://doi.org/10.16924/revinge.47.3>

Flores, A. (2021). Lista de las mejores librerías de Python para el 2022. [crehana.com. https://www.crehana.com/ec/blog/desarrollo-web/librerias-python/](https://www.crehana.com/ec/blog/desarrollo-web/librerias-python/)

Fuentes, M., & Medina, W. (2021). Diseño de un modelo predictivo-asistencial de pacientes infectados por Covid-19, mediante un modelo supervisado de Machine Learning basado en criterios de derivación hospitalaria o ambulatoria. Repositorio Universidad de Guayaquil. <http://repositorio.ug.edu.ec/bitstream/redug/52650/1/B-CISC-PTG-1901-2021%20Fuentes%20Marmolejo%20Melina%20Daniela%20-%20Medina%20Parra%20Wilmer%20David.pdf>

Landau, E. C., Guimaraes, D., & Hirsch, A. (2014). Uso de sistema de informaciones geográficas para espacialización de datos del área de producción agrícola. ResearchGate. https://www.researchgate.net/publication/280098376_Uso_de_Sistema_de_Informaciones_Geograficas_para_espacializacion_de_datos_del_area_de_produccion_agricola

Llamas, L., & Sulé, M. (2021). Tecnología en la agricultura. Innovación como método de evolución agrícola. Universidad de León. <https://buleria.unileon.es/>

[handle/10612/13619](https://doi.org/10.10612/13619)

Magaña, J. B. (2015). DETECCIÓN DE OBJETOS EN IMÁGENES UTILIZANDO OPENCV PARA RASPBERRY. ResearchGate. https://www.researchgate.net/publication/316282519_DETECCION_DE_OBJETOS_EN_IMAGENES_UTILIZANDO_OPENCV_PARA_RASPBERRY

Maniyath, S. R., Vinod, P., M, N., R, P., N, P. B., N, S., & Hebbar, R. (2018). Plant Disease Detection Using Machine Learning. International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 41-45. <https://doi.org/10.1109/icdi3c.2018.00017>

Marín, R. (2020). ¿Qué es OpenCV? Instalación en Python y ejemplos básicos. Canal Informática y TICS. <https://revistadigital.inesem.es/informatica-y-tics/opencv>

Núñez, O., Figueroa, T., & De Jesús, A. (2015). Monitorización de cultivos utilizando drones. Docplayer.es. <https://docplayer.es/13303546-Monitorizacion-de-cultivos-utilizando-drones-clave-del-proyecto-cin2015a20121.html>

Olafenwa, M. (2022). ImageAI: A Python library built to empower developers to build applications and systems with self-contained computer vision capabilities [Software]. En GitHub (2.1.6). <https://github.com/OlafenwaMoses/ImageAI>

Otálora, P., Sánchez, J., Fernández, F., & Soria, M. B. (2021). Herramienta gráfica para la caracterización de cultivos de microalgas basada en redes neuronales artificiales. En *Servizo de Publicacións da UDC eBooks* (pp. 119-125). <https://doi.org/10.17979/spudc.9788497498043.119>

Pusiol, P. (2014). Redes Convolucionales en Comprensión de Escenas. Repositorio Digital de la Universidad Nacional de Córdoba (RDU). <https://rdu.unc.edu.ar/bitstream/handle/11086/2799/TE%20C%20Pusiol.pdf?sequence=1&isAllowed=y>

Rojas, M. (2020). Machine Learning: Análisis

de lenguajes de Programación y herramientas para desarrollo- ProQuest. Proquest.com. <https://www.proquest.com/docview/2388304894?pq-origsite=gscholar&fromopenview=true>

Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019). A Review on Machine Learning Classification Techniques for Plant Disease Detection. 5th International Conference on Advanced Computing & Communication Systems, (ICACCS), 281-284. <https://doi.org/10.1109/icaccs.2019.8728415>

Sierra, J. (2022). Introducción a las redes neuronales artificiales. Repositorio Docta de la Universidad Complutense de Madrid. <https://hdl.handle.net/20.500.14352/3133>

Sotomayor, J. F., Gómez, A. P., & Cela, A. F. (2014). Sistema de Visión Artificial para el Análisis de Imágenes de Cultivo basado en Texturas Orientadas. Revista Politécnica, 33(1). https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/104

Sucar, L., & Gómez, G. (2017). Visión computacional. Instituto Nacional de Astrofísica, Óptica y Electrónica. México. <https://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>

Taquía-Gutiérrez, J. A. (2017). El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital. Interfases. <https://doi.org/10.26439/interfases2017.n10.1767>

Tanner, G. (2019). Introduction to Deep Learning with Keras. How to use the Keras Deep Learning library. <https://towardsdatascience.com/introduction-to-deep-learning-with-keras-17c09e4f0eb2>

Turned, T. (2017). What is TensorFlow? <https://medium.com/@deshayk/what-is-tensorflow-66ef027e5c6d>

Ultralytics. (2022). GitHub - Ultralytics/yolov3: YOLOV3 in PyTorch; ONNX; CoreML; TFLite. GitHub. <https://github.com/ultralytics/yolov3>