

Desarrollo de un algoritmo de visión artificial basado en redes neuronales para la identificación del oídium en la planta de arándano

Development of an artificial vision algorithm based on neural networks for the identification of oidium in the blueberry plant

Desenvolvimento de um algoritmo de visão computacional baseado em redes neurais para a identificação do oídio na planta de mirtilo

Bruno Crispin, Shelleny Lourdes¹

<https://orcid.org/0000-0002-0628-0686>

Ravines Robles, Victor Eduardo²

<https://orcid.org/0000-0002-8909-7878>

Santisteban León, Andrea Elizabeth³

<https://orcid.org/0000-0003-4696-5126>

Mendoza López, Rene Pedro⁴

<https://orcid.org/0000-0001-8326-5773>

Recibido: 10.08.2023

Aceptado: 06.11.2023

RESUMEN

En este estudio, introducimos un enfoque revolucionario que combina la visión por computadora y las redes neuronales con la finalidad de diseñar un algoritmo de visión por computadora que pueda distinguir entre las hojas de arándanos que presenten oidiosis y las que se encuentren sanas sin esta enfermedad, lo que podría mejorar significativamente la calidad y eficiencia de la producción de arándanos en la industria agroindustrial. Metodología. Se utilizó el lenguaje de programación Python y la biblioteca YOLOv7, se ejecutó en un procesador Intel Core i5, 8 GB de RAM, tarjeta gráfica Nvidia Geforce 1650, webcam con resolución 1080p. Mediante un enfoque experimental y cuantitativo, evaluamos la eficacia del algoritmo. Resultados. Se evidenció una eficiencia del 91% en la detección del hongo Oidio u Oidiosis en los arándanos, valida la efectividad del método y resalta su potencial para transformar la industria agroindustrial. Conclusión. Permitirá una clasificación más eficiente de los productos, lo que podría tener un impacto positivo en la calidad y competitividad del mercado de arándanos.

Palabras claves: enfermedad oidiosis, programación Python, visión artificial y redes neuronales.

ABSTRACT

In this study, we introduce a revolutionary approach that combines computer vision and neural networks in order to design a computer vision algorithm that can distinguish between blueberry leaves that have powdery mildew and those that are healthy without this disease. which could significantly improve the quality and efficiency of blueberry production in the agroindustrial industry. Methodology. The Python programming language and the YOLOv7 library were used, it was executed on an Intel Core i5 processor, 8 GB of RAM, Nvidia Geforce 1650 graphics card, webcam with 1080p resolution. Using an experimental and quantitative approach, we evaluate the effectiveness of the algorithm. Results. An efficiency of 91% was demonstrated in the detection of the Oidio or Oidiosis fungus in blueberries, validating the effectiveness of the method and highlighting its potential to transform the agro-industrial industry. Conclusion. It will allow for more efficient classification of products, which could have a positive impact on the quality and competitiveness of the blueberry market.

Keywords: powdery mildew disease, Python programming, computer vision and neural networks.

RESUMO

Neste estudo, apresentamos uma abordagem revolucionária que combina visão computacional e redes neurais para projetar um algoritmo de visão computacional que possa distinguir entre folhas de mirtilo que apresentam oídio e

¹ Fundo Montoro S.R.L. Chimbote. Perú. Ingeniero. shellenybruno1@gmail.com .

² Fundo Montoro S.R.L. Chimbote. Perú. Ingeniero. vravinesro13@gmail.com

³ Química Suiza S.A.C. Chorrillos. Perú. Ingeniero. andreasantisteban4@gmail.com .

⁴ Instituto Superior Tecnológico Público de Huarmey. Huarmey. Perú. Magister. rene.mendoza@icstphuarmey.edu.pe.

aquelas que são saudáveis sem esta doença, o que poderia melhorar significativamente a qualidade e eficiência da produção de mirtilo na indústria agroindustrial. Metodologia. Foi utilizada a linguagem de programação Python e a biblioteca YOLOv7, executada em processador Intel Corei5, 8 GB de RAM, placa gráfica Nvidia Geforce 1650, webcam com resolução 1080p. Utilizando uma abordagem experimental e quantitativa, avaliamos a eficácia do algoritmo. Resultados. Foi demonstrada uma eficiência de 91% na detecção do fungo Oídio ou Oidiose em mirtilos, validando a eficácia do método e destacando seu potencial para transformar a indústria agroindustrial. Conclusão. Permitirá uma classificação mais eficiente dos produtos, o que poderá ter um impacto positivo na qualidade e na competitividade do mercado do mirtilo.

Palavras Chave: *doença do oídio, programação Python, visão computacional e redes neurais.*

Introducción

Nuestro país, al ser un territorio biodiverso, siempre se ha caracterizado por tener la capacidad de generar múltiples cultivos y en grandes cantidades. Sin embargo, en todo cultivo existen plagas y/o enfermedades las cuales si no se detectan a tiempo llegan a afectar la salud de la planta. Según el Instituto Nacional de Estadística e Informática (INEI), en noviembre de 2022 se produjeron 42 mil 40 toneladas de arándanos, lo que representa un incremento del 16% con respecto al mismo mes del año anterior (Instituto Nacional de Estadística e Informática, 2022). El mercado internacional de arándanos es muy competitivo y exige altos estándares de calidad y sanidad. El país ha logrado posicionarse como uno de los principales exportadores de esta fruta en el mundo, superando a países como México, que también ha aumentado su producción en los últimos cinco años (Blue Berries Consulting, 2021). Según la Asociación de Productores y Exportadores de Arándanos del Perú (Proarándanos), las exportaciones alcanzaron los 874 millones de dólares en 2021, lo que representa un crecimiento del 23% con respecto al año anterior (Portal Frutícola, 2022).

El oídio u oidiosis es una enfermedad fúngica que afecta a las plantas de arándano principalmente en el follaje perjudicando a la calidad de la planta (Agencia Agraria de Noticias, 2022). Esta enfermedad puede reducir la calidad de la fruta y disminuir los rendimientos de la cosecha. La resistencia al oídio varía entre las diferentes variedades de arándano (Fernández, 2018) y se están realizando investigaciones para desarrollar variedades más resistentes. El manejo del oídio en arándanos implica una combinación de prácticas culturales, control biológico y, si es necesario, control químico (Patogenicidad y control químico de oídio en arándano en el sur de Chile). A pesar de estos desafíos, los arándanos siguen siendo una fruta popular y valiosa en la agricultura. Entre las medidas de control de la enfermedad se encuentra la higiene dentro del área de cultivo, proporcionar una ventilación adecuada, evitar el exceso de humedad (Fernandez, 2023) o el control del nitrógeno evitando el uso excesivo de abonos con alto contenido de nitrógeno (Juste, 2022). Otro de los métodos es la aplicación de pulverizaciones de azufre espolvoreando las plantas afectadas, las cercanas y el medio para eliminar el oídio (Gras, 2023).

Las plantas de arándano presentan los siguientes signos cuando se ven afectadas por el oídio como la aparición de una capa blanca de aspecto algodonoso, hojas amarillas y caídas, y una coloración rojiza en las bayas verdes (De Souza Pacheco, 2021). Ante esto surge como alternativa, la utilización de la tecnología mediante la elaboración de un algoritmo que permita identificar con mayor rapidez y en tiempo real la presencia del hongo en la planta de arándano usando en su desarrollo técnicas de visión artificial y redes neuronales convolucionales. Las redes neuronales son esencialmente una representación artificial y simplificada del cerebro humano, capaz de adquirir conocimiento a través de la experiencia, las redes neuronales convolucionales (CNN) son arquitecturas especializadas en el procesamiento de datos bidimensionales, como imágenes o vídeos (Matich, 2001). Utilizan capas de convolución para identificar patrones visuales y capas de pooling que reducen la dimensionalidad de la información. Esta combinación permite a las CNN aprender jerarquías de características cada vez más abstractas, siendo efectivas en tareas de reconocimiento visual, desde la clasificación de imágenes hasta aplicaciones en medicina y conducción autónoma (Krizhevsky et al., 2017) y (Jimenez, 2021). En este proyecto, se emplea Python debido a su capacidad de funcionar en diversas plataformas (Mac, Linux, Windows) y su disponibilidad de bibliotecas especializadas, así como herramientas para el desarrollo de redes neuronales artificiales, incluyendo las convolucionales.

Material y métodos:

Para la implementación del sistema de visión artificial se seleccionó como lenguaje de programación el Python en su versión 3.11, cuenta con bibliotecas de visión artificial como OpenCV, facilidad de uso por su sintaxis, integración con bibliotecas como Numpy y la visualización de datos (Del Valle Hernandez, 2019), (Hualde, 2019) y (Structuralia, 2020), para las pruebas se utilizó un ordenador portátil Lenovo Legion con procesador Intel Corei5 10ma generación, 8 GB de RAM y potenciado por una tarjeta gráfica Nvidia GeForce GTX 1650 trabaja en conjunto con CUDA para la aceleración de las aplicaciones de cálculo intensivo (HardZone, 2023). Adicionalmente se construyó un ambiente controlado con un foco LED de 5 watt el cual funciona como fuente de iluminación, para la captura de las imágenes se ha colocado una webcam vía USB la cual asegura que las imágenes captadas puedan contar con las características adecuadas, así como se muestra en la Figura 1.

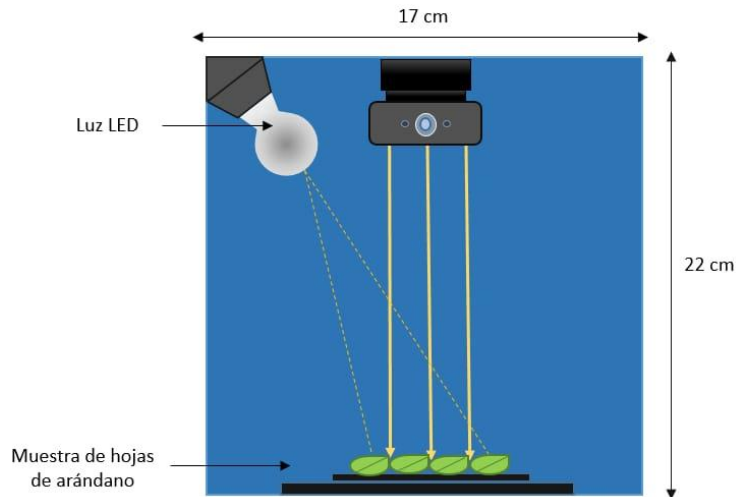


Figura 1. Ambiente controlado

El preprocesamiento: Un procesamiento previo que se aplica a las imágenes digitales con el fin de eliminar la mayor cantidad de ruido dentro del modelo para lograr maximizar las características propias. Como tercera etapa tenemos la segmentación: Proceso mediante el cual la imagen se divide en varias partes, a cada píxel se le asigna una categoría y estas varían según la segmentación elegida. Extracción de características: En este proceso, a partir de la imagen obtenemos características significativas, se reduce el conjunto de datos sin procesar a características más manejables, facilitando el aprendizaje y generalización de los modelos y reduciendo los recursos utilizados para el procesamiento de la información. Finalmente, la clasificación y reconocimiento: Es el paso en donde se predice la etiqueta de las imágenes con ayuda del modelo elegido, se reconoce el objeto, las características y patrones para luego asignarle la etiqueta. el objetivo primordial es reconocer las características que presenta una planta de arándano que sí se ve afectada por el oídio en comparación con una que no cuenta con ningún problema, es de vital importancia que las fotografías obtenidas serán de una buena calidad para que los resultados no se vean afectados, ante ello se plantearon como parámetros principales de evaluación y control para la calidad de las fotografías el uso de un celular de última generación para que pueda realizar las fotografías adecuadas y una buena iluminación para la identificación de la presencia de oídio. Se obtuvieron 410 fotografías en diferentes ángulos, de las cuales 329 reflejaban el daño ocasionado por oídio y los 81 restantes mostraban a la hoja sana.

El dataset fue generado en base a una serie de 410 imágenes correspondientes a hojas de la planta de arándano, las cuales fueron clasificadas como “hoja con oidiosis” y “hoja sana sin oidiosis”. Además, para asegurar que el dataset sea el adecuado es importante que las imágenes se encuentren normalizadas, es decir, deben tener las mismas dimensiones y tener el mismo tamaño. La Figura 2 muestra la hoja de la planta de arándano cuando se encuentra totalmente sana y cuando esta tiene presencia de oidiosis.

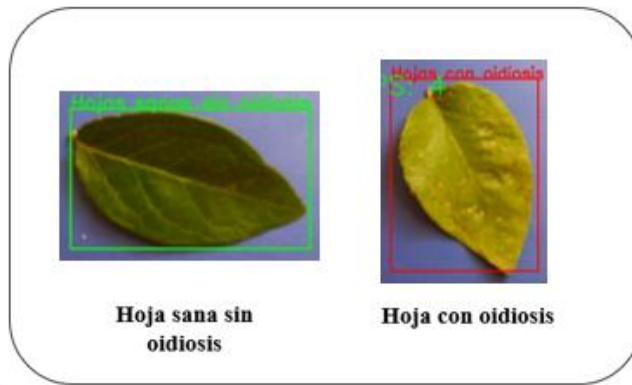


Figura 2. Clasificación de hoja sana y hoja con oidiosis

Luego, las imágenes recolectadas pasan al proceso de etiquetado, estas etiquetas presentan un formato determinado para la red neuronal que se está utilizando (YoloV7) el cual se muestra en la Figura 3.

class/x/y/width/height

Figura 3. Formato de etiquetado.

Para etiquetar las imágenes se utilizó el etiquetador de Roboflow, el cual es una herramienta que permite etiquetar imágenes de manera rápida y sencilla. En la Figura 4 se muestra resultados del etiquetado de una de las muestras en el formato determinado. Cumpliendo así con los datos de la Figura 3.

0	0.2296875	0.1390625	0.1734375	0.20625
0	0.3875	0.3328125	0.1859375	0.190625
0	0.28359375	0.475	0.203125	0.1703125
0	0.66484375	0.51796875	0.19921875	0.1984375
0	0.49921875	0.728125	0.3375	0.1796875

Figura 4. Etiqueta de una muestra.

Para la etapa de entrenamiento de la red, se hizo uso de una técnica llamada Transfer Learning, la cual consiste en utilizar un modelo de IA ya entrenado y modificarlo para que se adapte a las necesidades del proyecto. Esta técnica es muy utilizada para desarrollar modelos eficaces y capaces de resolver problemas complejos (Data Scientist, 2023).

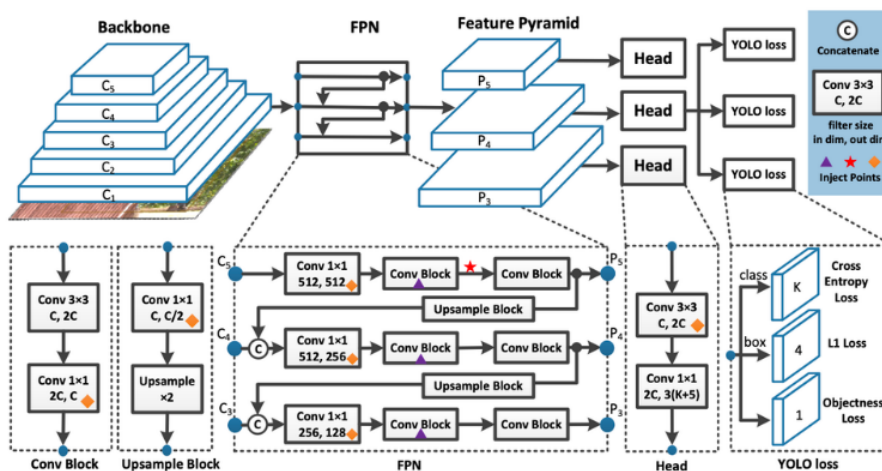


Figura 5. Arquitectura YOLOv7.

El modelo de IA seleccionado para utilizar la técnica ya mencionada se llama YOLOv7, el cual es un modelo de IA que se utiliza, para la detección de objetos en imágenes. Este se basa en redes neuronales convolucionales, las cuales son un tipo de red neuronal que se utiliza para la clasificación de imágenes. El algoritmo YOLO utiliza una única red neuronal convolucional para detectar objetos en imágenes. La red neuronal divide la imagen en regiones, prediciendo cuadros de identificación y probabilidades por cada región; las cajas son ponderadas a partir de las probabilidades predichas. La red tiene 24 capas convolucionales seguidas por 2 capas de conexión completa. Esta hace uso de capas de reducción de 1x1 seguidas capas convolucionales de 3x3. La salida final del modelo tensor de predicción de 7x7x30. La arquitectura de YOLOv7 se puede apreciar en la Figura 5.

Para la etapa de entrenamiento se usa el comando mostrado en la Figura 6.

```
python .\yolov7\train.py --workers 1 --device 0
--batch-size 4 --data data.yml --img 640 640 --cfg .
\yolov7\cfg\training\yolov7.yaml --weights 'yolov7.pt'
--name avocado --hyp
.\yolov7\data\hyp.scratch.p5.yaml --epochs 4
```

Figura 6. Comando de entrenamiento de red neuronal

Los workers son los núcleos de la computadora que se van a utilizar para entrenar el modelo. El device es el número de la tarjeta gráfica que se va a utilizar para entrenar el modelo. El batch-size es el número de imágenes que se van a utilizar para entrenar el modelo. El data es el archivo de configuración de los datos. El img es el tamaño de las imágenes que se van a utilizar para entrenar el modelo. El cfg es el archivo de configuración del modelo. El weights es el modelo que se va a utilizar para entrenar el modelo. El name es el nombre del modelo. El hyp es el archivo de configuración de los hiperparámetros. El epochs es el número de veces que se va a entrenar el modelo.

Inicialización. Para inicializar el algoritmo se usa un método de inicialización llamado constructor, el cual está preparando un modelo de aprendizaje automático para hacer predicciones, y decide si debe usar la CPU o una GPU (si está disponible). En la Figura 7, se aprecia el código utilizado.

```
self.model = self.load_model()
self.classes = self.model.names
self.device = 'cuda' if torch.cuda.is_available()
else 'cpu'
print("\n\nDevice Used:",self.device)
```

Figura 7. Comando de inicialización.

Carga del modelo de detección de objetos. Posteriormente se debe cargar el modelo de detección de objetos para el correcto funcionamiento del algoritmo el cual funciona bajo el siguiente código mostrado en la Figura 8.

```
model = torch.hub.load('WongKinYiu/yolov7',
'custom', 'model-oidiosis-v2.pt',
force_reload=False, trust_repo=True)
```

Figura 8. Comando de carga del modelo

Dibujo de cajas delimitadoras. Tomando las etiquetas y las coordenadas de los objetos detectados por el modelo en el frame, ahora el algoritmo se encargará de trazar las cajas delimitadoras y las etiquetas en el frame. Mediante el código mostrado en la Figura 9, se van a poder apreciar los labels al momento de ejecutar el algoritmo, facilitando así la visión y determinación de las hojas con oidiosis y las hojas sanas sin oidiosis, como en la Figura 10.

```
labels, cord = results
n = len(labels)
x_shape, y_shape = frame.shape[1],
frame.shape[0]
for i in range(n):
```

```

row = cord[i]
if row[4] >= 0.01:
    x1, y1, x2, y2 = int(row[0]*x_shape),
int(row[1]*y_shape), int(row[2]*x_shape),
int(row[3]*y_shape)
    if labels[i] == 0:
        bgr = (0, 0, 255)
    else:
        bgr = (0, 255, 0)
    cv2.rectangle(frame, (x1, y1), (x2, y2), bgr,
cv2.putText(frame,
self.class_to_label(labels[i]), (x1, y1),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, bgr, 2)

```

Figura 9. Comando plot_boxes



Figura 10. Display de los labels.

Resultados

Se implementó la tecnología de visión artificial para identificar la presencia del oidio en las hojas de arándano. Se realizaron 100 pruebas con hojas en diversas posiciones y tamaños para evaluar la precisión de la red neuronal, como se puede apreciar en la Figura 11 y Figura 12. Estas 100 pruebas fueron colocadas dentro del ambiente controlado de la Figura 1.

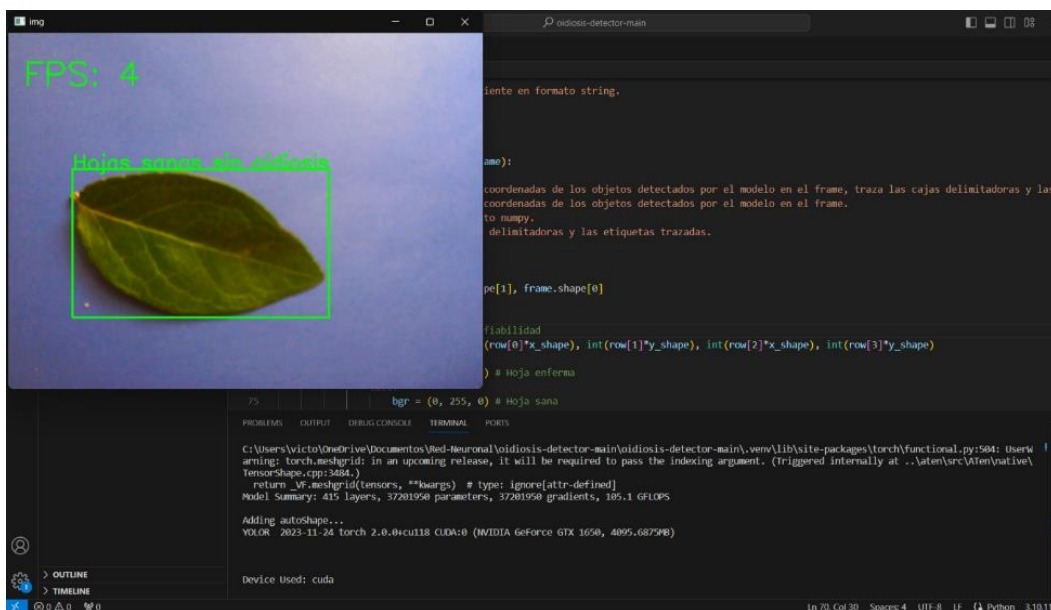


Figura 11. Hoja sana sin oidiosis

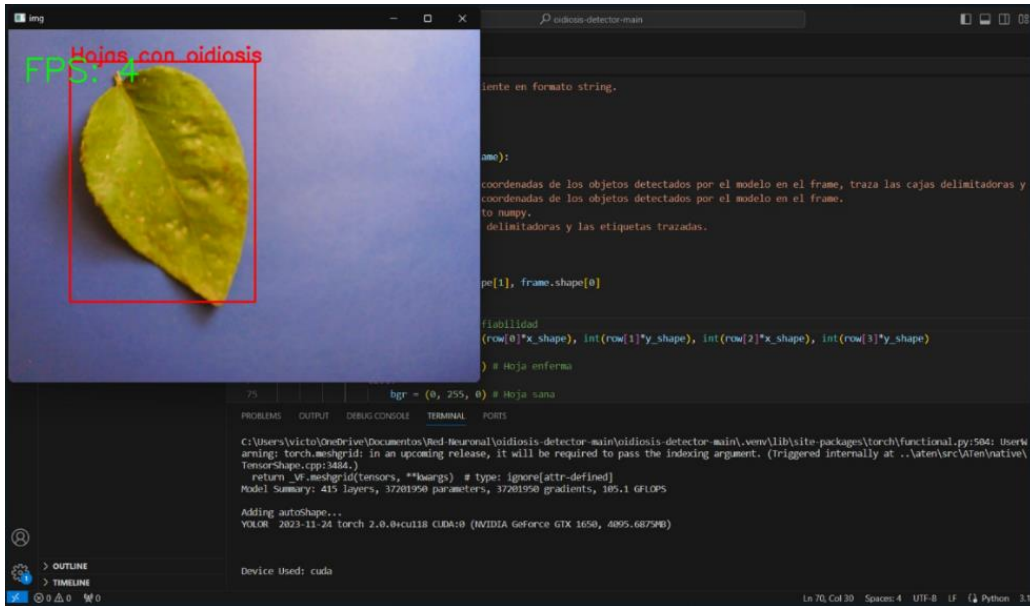


Figura 12. Hoja con oidiosis

Se determinó a su vez la precisión del algoritmo. Mediante los datos mostrados en la Tabla 1.

Tabla 1.

Resultados de las pruebas de precisión

Cantidad de pruebas	Datos reales	Datos obtenidos	Resultados	Cantidad de pruebas	Datos reales	Datos obtenidos	Resultados
1	sana	sana	VP	51	sana	sana	VP
2	oidio	oidio	VN	52	oidio	oidio	VN
3	sana	sana	VP	53	oidio	oidio	VN
4	sana	sana	VP	54	oidio	oidio	VN
5	oidio	oidio	VN	55	oidio	oidio	VN
6	sana	sana	VP	56	oidio	oidio	VN
7	oidio	oidio	VN	57	oidio	oidio	VN
8	sana	sana	VP	58	sana	sana	VP
9	oidio	oidio	VN	59	oidio	oidio	VN
10	sana	oidio	FN	60	sana	sana	VP
11	sana	sana	VP	61	sana	sana	VP
12	sana	sana	VP	62	sana	sana	VP
13	sana	sana	VP	63	sana	sana	VP
14	oidio	oidio	VN	64	sana	sana	VP
15	oidio	sana	FP	65	sana	oidio	FN
16	sana	sana	VP	66	sana	oidio	FN
17	sana	sana	VP	67	oidio	oidio	VN
18	sana	sana	VP	68	oidio	oidio	VN
19	sana	sana	VP	69	oidio	sana	FP
20	sana	sana	VP	70	oidio	oidio	VN
21	oidio	sana	FP	71	oidio	oidio	VN
22	oidio	oidio	VN	72	oidio	oidio	VN
23	sana	sana	VP	73	oidio	oidio	VN
24	sana	sana	VP	74	sana	sana	VP
25	oidio	oidio	VN	75	sana	sana	VP
26	oidio	oidio	VN	76	oidio	oidio	VN
27	oidio	oidio	VN	77	oidio	oidio	VN

28	oidio	oidio	VN	78	sana	sana	VP
29	oidio	oidio	VN	79	sana	sana	VP
30	oidio	sana	FP	80	sana	sana	VP
31	sana	sana	VP	81	oidio	oidio	VN
32	sana	sana	VP	82	oidio	oidio	VN
33	sana	sana	VP	83	oidio	sana	FP
34	sana	sana	VP	84	oidio	oidio	VN
35	sana	sana	VP	85	oidio	oidio	VN
36	sana	sana	VP	86	oidio	oidio	VN
37	oidio	oidio	VN	87	sana	sana	VP
38	oidio	oidio	VN	88	sana	sana	VP
39	oidio	oidio	VN	89	oidio	oidio	VN
40	oidio	oidio	VN	90	oidio	oidio	VN
41	sana	sana	VP	91	oidio	oidio	VN
42	sana	sana	VP	92	oidio	oidio	VN
43	sana	sana	VP	93	oidio	oidio	VN
44	sana	sana	VP	94	oidio	oidio	VN
45	oidio	oidio	VN	95	sana	sana	VP
46	sana	sana	VP	96	sana	sana	VP
47	sana	sana	VP	97	oidio	oidio	VN
48	sana	sana	VP	98	oidio	oidio	VN
49	sana	sana	VP	99	oidio	sana	FP
50	sana	sana	VP	100	oidio	oidio	VN

Matriz de confusión

		Datos Obtenidos	
		Positivo	Negativo
Datos Reales	Positivo	46	3
	Negativo	6	45

- a) Tras completar el análisis de la matriz de confusión, procedemos a utilizar la fórmula para calcular la exactitud, la cual nos indica qué tan cercanos estuvieron los datos reales a los datos obtenidos.

$$Accuracy = \frac{Predicciones\ correctas}{Total\ de\ predicciones}$$

$$Accuracy = \frac{46 + 45}{46 + 3 + 6 + 45}$$

$$Accuracy = 0.91$$

- b) La precisión se utiliza para determinar qué porcentaje de todas las pruebas realizadas, identificadas como positivas, realmente son casos positivos.

$$Precisión = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos}$$

$$Precisión = \frac{46}{46 + 3} \quad Precisión = 0.92$$

- c) El recall, una métrica de evaluación, se utiliza para determinar cuántos de los casos positivos reales son identificados correctamente.

$$Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}$$

$$Recall = \frac{50}{50 + 6} \quad Recall = 0.89$$

- d) *Por último*, el F1 score, una métrica de evaluación, combina las medidas de precisión y recall en una única puntuación. Si esta puntuación se acerca más a 1, indica que el programa está cumpliendo con su propósito.

$$F1\ Score = \frac{2}{\frac{1}{Precisión} + \frac{1}{Recall}}$$

$$F1\ Score = \frac{2}{\frac{1}{0.92} + \frac{1}{0.89}}$$

$$F1\ Score = \frac{2}{\frac{1}{0.92} + \frac{1}{0.89}} \quad F1\ Score = 0.90$$

Posteriormente, se realizaron las pruebas del funcionamiento del algoritmo en un ambiente no controlado, obteniendo los resultados:

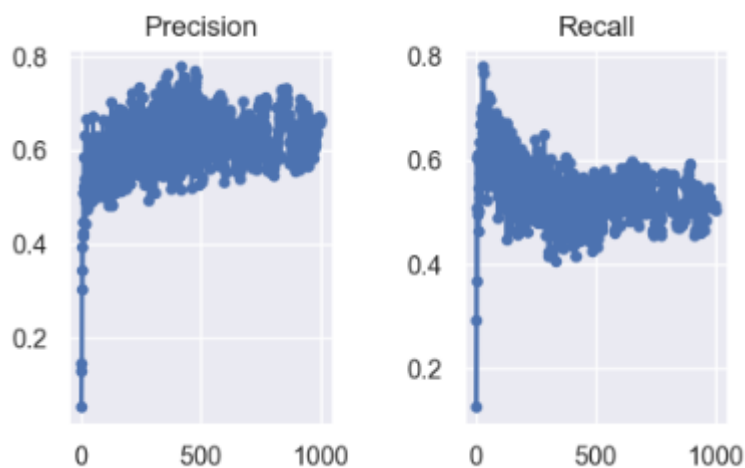


Figura 13. Precisión y recall del algoritmo en ambiente no controlado

Como se aprecia en la Figura 13 la precisión de estas pruebas no es estable, teniendo un pico de precisión del 80%, y un Recall de aproximadamente 50%, esto es debido a que el entrenamiento del algoritmo se realizó con fotografías en un ambiente controlado. Las pruebas realizadas en exteriores se muestran en la Figura 14.

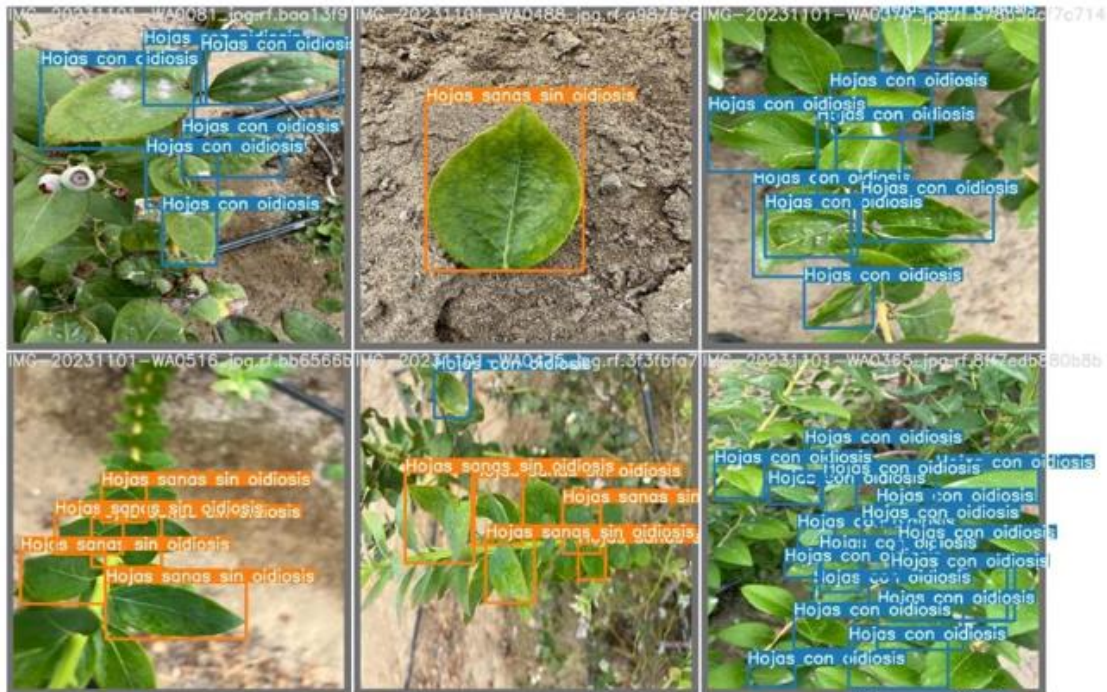


Figura 14. Resultados del algoritmo en ambiente no controlado

Discusión

Se utilizó tecnología de visión artificial para identificar hojas de la planta de arándano afectadas por el hongo. Durante este procedimiento, se llevaron a cabo un total de 100 pruebas, de las cuales 52 pruebas eran hojas sanas y 48 eran hojas con la enfermedad de oidiosis, obteniendo una exactitud del 91%. Estas pruebas se realizaron con diversas proporciones y cantidades de hojas con el objetivo de evaluar la eficacia de la red neuronal. Utilizando el reconocimiento de redes neuronales, se realiza la identificación de la categoría a la que pertenece cada hoja y se marca con colores para distinguir entre hojas defectuosas y sanas. La investigación ha conseguido crear un algoritmo de visión artificial que permite la detección automatizada del hongo *Microsphaera vaccinii* o *Erysiphe vaccinii*, comúnmente llamada “oídio/oidiosis” en las hojas de la planta de arándanos. Durante el estudio, se contrastaron los resultados obtenidos con investigaciones previas que emplearon distintos enfoques y tecnologías para abordar problemas similares.

En el estudio relacionado, se utilizó la arquitectura VGG19 para la detección de la enfermedad *Phytophthora infestans* en papa *Solanum Tuberosu*. Esta arquitectura logró una eficiencia del 79% en la detección de la enfermedad, se utilizaron un conjunto de 120 imágenes de prueba (Lozada-Portilla et al., 2021) En comparación, el presente trabajo utilizó la arquitectura YOLOv7 y una red neuronal basada en visión artificial para identificar las hojas de la planta de arándano que tengan la enfermedad de oídio y las que se encuentren sanas, alcanzando una eficiencia del 91% con un conjunto de 100 imágenes de prueba. Esto indica un notable avance en la exactitud de la detección en contraste con la investigación previa.

También, otra investigación vinculada presentó un algoritmo detector de nivel de clorofila tipo A y B en hojas a partir de una red neuronal convolucional. Este método híbrido utilizó 250 imágenes para la fase de entrenamiento, teniendo una precisión del 98,22%, además se menciona que, para tener una adecuada precisión se debe entrenar al algoritmo con una cantidad de muestras de 1000 imágenes (Espinosa y Gomez, 2020). En comparación con el presente trabajo, se utilizaron 410 imágenes para la fase de entrenamiento, de las cuales 329 imágenes presentaban el hongo oídio en las hojas de arándano y 81 imágenes presentaban hojas sanas, obteniendo una precisión del 92%. Esto indica que, de haber realizado el entrenamiento del algoritmo con una muestra no menor de 1000 imágenes, se hubiera obtenido una precisión más elevada.

Conclusiones

Se concluye primero que, de acuerdo al objetivo principal de la investigación, la cual fue desarrollar un algoritmo de visión artificial que permita clasificar que hojas de la planta de arándano se encuentran sanas y cuales se encuentran enfermas con oidiosis, lo cual se logró completar de manera satisfactoria.

Este algoritmo permite clasificar exitosamente los dos tipos de hojas. Puesto que el software generado nos permitió realizar dicha clasificación con un nivel de precisión y de recall del 90%.

Referencias

Agencia Agraria de Noticias. (2022). Oidiosis, un nuevo reto a controlar en el cultivo de arándano. [Reportaje] 23 de febrero de 2022.

Blue Berries Consulting. (2021). Producción de Arándanos en México crece 5 años seguidos. [Noticia] 26 de noviembre de 2021.

Data Scientist. (2023). ¿Qué es el Transfer Learning? *DataScientest.com*. Data Science, 2023. <https://datascientest.com/es/que-es-el-transfer-learning>

De Souza Pacheco, Joao. (2021). La oidiosis del arándano *Microsphaera vaccinii*. *Ecofertilizing*, 2 de septiembre de 2021.

Del Valle Hernandez, Luis. (2019). Visión artificial, OpenCV y Python. Primeros pasos para analizar imágenes. *Programarfacil.com*. <https://programarfacil.com/podcast/81-vision-artificial-opencv-phyton/>

Espinosa, M. A., & Gomez, A. F. (2020). Diseño de un algoritmo detector de nivel de clorofila tipo A y B en las hojas mediante procesamiento de imágenes aplicando una red neuronal convolucional. <http://repositorio.uts.edu.co:8080/xmlui/handle/123456789/2231>

Fernández, J. J. F., Madrera, R. R., & Negrillo, A. M. C. (2018). Fenotipado de la colección de arándanos del SERIDA mediante análisis de imágenes de frutos. In *IX Congreso de Mejora Genética de Plantas: Murcia 2018* (pp. 252-255). Instituto Murciano de Investigación y Desarrollo Agrario y Alimentario.

Fernandez, Lucia. (2023). Oídio: El Enemigo Silencioso de Tus cultivos y Cómo Combatirlo. [Noticia] s.l. : Agriculteca, 16 de Mayo de 2023.

Gras, Manuel. (2023). Oídio en plantas, qué es, identificación y tratamiento. s.l. : Jardínatis, 13 de Noviembre de 2023.

HardZone. (2023). Tu gráfica nvidia tiene cientos, ¿Sabes qué son los núcleos Cuda? *HardZone*. [En línea] 29 de noviembre de 2023. <https://hardzone.es/marcas/nvidia/nucleos-cuda/>

Hualde. (2019). Introducción a la Visión Artificial Con opencv y python. [En línea] 2019. <https://rostutorial.com/vision-artificial-intro/>.

Instituto Nacional de Estadística e Informática. (2022). Producción de Arándano Alcanzó 42 mil 40 toneladas en noviembre del año 2022. [Noticia] s.l. : Plataforma del Estado Peruano, 2022.

Jimenez, M. (2021) Clasificación de piezas metálicas mediante una Red Neuronal Convolutiva y Análisis de Componentes Principales. <http://hdl.handle.net/20.500.11799/137394>

Juste, Irene. (2022). *Cómo combatir El Oídio con Remedios Caseros - Los Métodos Más Efectivos*. s.l. : Ecología Verde, 01 de Julio de 2022.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90. <https://doi.org/10.1145/3065386>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. <https://doi.org/10.1109/5.726791>

Lozada-Portilla, W. A., Suarez-Barón, M. J., & Avendaño-Fernández, E. (2021). Aplicación de redes neuronales convolucionales para la detección del tizón tardío *Phytophthora infestans* en papa *Solanum tuberosum*. *Revista UDCA Actualidad & Divulgación Científica*, 24(2). <https://doi.org/10.31910/rudca.v24.n2.2021.1917>

Matich, D. J. & Ruiz, C. A., Basualdo, M. S. (2001). Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional–Facultad Regional Rosario-Departamento de Ingeniería Química-Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ)*, 41, 12-16.

Portal Fruticola. (2022). *Proarándanos: “Las exportaciones peruanas de arándanos crecieron en volumen más de 16 veces”*. [Noticia] 23 de marzo de 2022.

Structuralia. 2020. La utilidad de Python para la inteligencia artificial. *Structuralia*. [En línea] 23 de Octubre de 2020. <https://blog.structuralia.com/python-inteligencia-artificial>.