

# Diego Rengifo

## Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en i...

 Revisión de Tesis final - Unidad de Investigación FISI

---

### Detalles del documento

Identificador de la entrega

trn:oid:::3117:581310812

Fecha de entrega

21 abr 2026, 8:45 GMT-5

Fecha de descarga

21 abr 2026, 8:54 GMT-5

Nombre del archivo

(Tesis) DIEGO (1).pdf

Tamaño del archivo

1.2 MB

79 páginas

19.583 palabras

114.788 caracteres

# 12% Similitud general




El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

## Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto citado
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 10 palabras)

---

## Fuentes principales

- 9%  Fuentes de Internet
- 2%  Publicaciones
- 9%  Trabajos entregados (trabajos del estudiante)

---

## Marcas de integridad

N.º de alertas de integridad para revisión

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

## Fuentes principales

- 9% Fuentes de Internet
- 2% Publicaciones
- 9% Trabajos entregados (trabajos del estudiante)

## Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	Internet	repositorio.unsm.edu.pe	4%
2	Internet	tesis.unsm.edu.pe	2%
3	Trabajos del estudiante	Universidad Nacional de San Martín on 2024-03-19	<1%
4	Internet	repositorio.upt.edu.pe	<1%
5	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-17	<1%
6	Trabajos del estudiante	Universidad Politecnica Salesiana del Ecuador on 2024-03-07	<1%
7	Internet	hdl.handle.net	<1%
8	Trabajos del estudiante	Universidad Nacional de San Martín on 2026-03-06	<1%
9	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-05	<1%
10	Trabajos del estudiante	Munster Technological University (MTU) on 2025-06-12	<1%
11	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-06	<1%

12	Trabajos del estudiante	Universidad Internacional de la Rioja on 2025-06-09	<1%
13	Internet	revistas.uclave.org	<1%
14	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-06	<1%
15	Trabajos del estudiante	Universidad Francisco de Vitoria on 2025-06-02	<1%
16	Trabajos del estudiante	Universidad Internacional de la Rioja on 2025-10-22	<1%
17	Internet	lmsspada.kemdiktisaintek.go.id	<1%
18	Trabajos del estudiante	Consortio CIXUG on 2024-08-26	<1%
19	Trabajos del estudiante	Universidad Estatal Amazonica- on 2025-10-29	<1%
20	Trabajos del estudiante	Universidad Nacional de San Martín on 2022-07-25	<1%
21	Trabajos del estudiante	Universidad Ricardo Palma on 2024-10-06	<1%
22	Trabajos del estudiante	Universidad Nacional de San Martín on 2024-01-06	<1%
23	Trabajos del estudiante	Universidad Nacional de San Martín on 2024-01-07	<1%
24	Trabajos del estudiante	Universidad de Guadalajara - Posgrados on 2025-12-03	<1%
25	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-07	<1%

26	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-07	<1%
27	Publicación	"Intelligent and Fuzzy Systems", Springer Science and Business Media LLC, 2025	<1%
28	Trabajos del estudiante	Infile on 2022-12-31	<1%
29	Trabajos del estudiante	Universidad Internacional de la Rioja on 2025-05-02	<1%
30	Trabajos del estudiante	Universidad Nacional de San Martín on 2024-01-08	<1%
31	Trabajos del estudiante	Universidad Rey Juan Carlos on 2026-04-08	<1%
32	Internet	itconnect.lat	<1%
33	Internet	joaquinamatrodrigo.github.io	<1%
34	Trabajos del estudiante	Corporación Universitaria Minuto de Dios, UNIMINUTO on 2024-08-13	<1%
35	Trabajos del estudiante	Pontificia Universidad Catolica del Ecuador - PUCE on 2024-06-20	<1%
36	Trabajos del estudiante	UNIBA on 2025-11-10	<1%
37	Trabajos del estudiante	Universidad Carlos III de Madrid - EUR on 2024-06-12	<1%
38	Trabajos del estudiante	Universidad Nacional de San Martín on 2025-12-15	<1%
39	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-03-06	<1%

40	Publicación	Yildirim, Salih. "Cografya OgretmenlerIne C-tpab ModelI TemellI ACik Kaynak Kod..."	<1%
41	Internet	boletin.invemar.org.co	<1%
42	Publicación	Paul, Prosenjit. "Towards Secure Autonomous Vehicles: An Integrated Edge and ..."	<1%
43	Trabajos del estudiante	Universidad Internacional de la Rioja on 2025-01-15	<1%
44	Trabajos del estudiante	Universidad Nacional de San Martín on 2022-09-20	<1%
45	Trabajos del estudiante	Universidad Tecnológica Centroamericana UNITEC on 2024-03-02	<1%
46	Trabajos del estudiante	Universitat Oberta de Catalunya on 2026-04-14	<1%
47	Internet	dione.lib.unipi.gr	<1%
48	Internet	documents.mx	<1%
49	Internet	patents.google.com	<1%
50	Internet	riull.ull.es	<1%
51	Internet	www.yumpu.com	<1%
52	Trabajos del estudiante	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2020-01-27	<1%
53	Trabajos del estudiante	Corporación Universitaria Minuto de Dios, UNIMINUTO on 2024-04-27	<1%

54	Trabajos del estudiante	Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López on 2025...	<1%
55	Trabajos del estudiante	FUNIBER on 2026-03-29	<1%
56	Trabajos del estudiante	UNIBA on 2024-11-13	<1%
57	Trabajos del estudiante	Universidad Nacional de San Martín on 2022-04-04	<1%
58	Trabajos del estudiante	Universidad Nacional de San Martín on 2026-02-24	<1%
59	Trabajos del estudiante	Universidad de Guayaquil on 2026-02-21	<1%
60	Trabajos del estudiante	Unviersidad de Granada on 2017-02-21	<1%
61	Internet	doi.org	<1%
62	Internet	doku.pub	<1%
63	Internet	repositorio.ucv.edu.pe	<1%
64	Internet	ri.uaemex.mx	<1%
65	Internet	www.unab.edu.co	<1%



Esta obra está bajo una

[Licencia Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Atribución - 4.0 Internacional \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Vea una copia de esta licencia en

<https://creativecommons.org/licenses/by/4.0/deed.es>





**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

Tesis

# **Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao**

Para optar el título profesional de Ingeniero de Sistemas e Informática

**Autor:**

Diego Emmanuel Rengifo Lozano

<https://orcid.org/0009-0008-8868-3331>

**Asesor:**

Ing. MBA. John Clark Santa Maria Pinedo

<https://orcid.org/0000-0002-8594-4865>

**Tarapoto, Perú**

**2026**



**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

## Tesis

# Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao

1 Para optar el título profesional de Ingeniero de Sistemas e Informática

### Autor:

Diego Emmanuel Rengifo Lozano

22 Sustentado y aprobado el 27 de marzo del 2026, ante el honorable jurado:

---

#### Presidente de Jurado

Lic. Dr. Edwin Augusto  
Hernández Torres

---

#### Secretario de Jurado

Ing. M. Sc. Pamela Magnolia  
Granda Milón

---

#### Vocal de Jurado

Ing. M. Sc. Juan Francisco  
Agreda Vega

---

#### Asesor

Ing. MBA. John Clark Santa  
Maria Pinedo

1 **Tarapoto, Perú**

**2026**

## Constancia de asesoramiento

Quien suscribe el presente documento,

### HACE CONSTAR:

Que, habiendo acompañado en la ejecución de la tesis titulada: Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao. Elaborado por el tesista:

Bachiller en Ingeniería de Sistemas e Informática: **Diego Emmanuel Rengifo Lozano**

Por lo que doy conformidad para los trámites correspondientes, dejo como constancia el presente documento y firmo.

Tarapoto, 27 de marzo de 2026

.....  
**Ing. MBA. John Clark Santa Maria Pinedo**  
Asesor

## Declaratoria de Autenticidad

2

Yo, Diego Emmanuel Rengifo Lozano, con DNI N° 72803521, bachiller de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, autor de la tesis titulada: Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao.

1

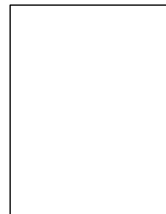
Declaramos bajo juramento que:

1. La tesis presentada es de autoría propia.
2. La redacción fue realizada respetando las citas y referencia de las fuentes bibliográficas consultadas, siguiendo las normas APA actuales.
3. Toda información que contiene la tesis no ha sido plagiada.
4. Los datos presentados en los resultados son reales, no han sido alterados ni copiados, por tanto, la información de esta investigación debe considerarse como aporte a la realidad investigada.

Por lo antes mencionado, asumo bajo responsabilidad las consecuencias que deriven de mi accionar, sometiéndome a las leyes de nuestro país y normas vigentes de la Universidad Nacional de San Martín.

Tarapoto, 27 de marzo de 2026.

.....  
**Diego Emmanuel Rengifo Lozano**  
DNI N° 72803521



## Ficha de identificación

<p><b>Título:</b> Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao</p>	<p><b>Área de investigación:</b> Ingeniería y Tecnología  <b>Línea de investigación:</b> Estrategias de tecnologías de información y comunicación (TIC) y sistemas constructivos convencionales y no convencionales para el desarrollo sostenible.  <b>Sub línea de investigación:</b> Inteligencia artificial y recuperación de información.  <b>Grupo de investigación:</b> Grupo de Investigación de IA (RESOLUCIÓN N° 134–2021–UNSM/FISI/CFT).  <b>Tipo de investigación:</b>                      Básica <input type="checkbox"/>, Aplicada <input checked="" type="checkbox"/>, Desarrollo experimental <input type="checkbox"/></p>
--	--

<p><b>Autor:</b> Diego Emmanuel Rengifo Lozano</p>	<p>Facultad de Ingeniería de Sistemas e Informática                      Escuela Profesional de Ingeniería de Sistemas e Informática  <a href="https://orcid.org/0009-0008-8868-3331">https://orcid.org/0009-0008-8868-3331</a></p>
--	---

<p><b>Asesor:</b> Ing. MBA. John Clark Santa María Pinedo</p>	<p>Dependencia local de soporte:                      Facultad de Ingeniería de Sistemas e Informática                      Escuela Profesional de Ingeniería de Sistemas e Informática                      Unidad o Laboratorio Ingeniería de Sistemas e Informática  <a href="https://orcid.org/0000-0002-8594-4865">https://orcid.org/0000-0002-8594-4865</a></p>
---	---

## Dedicatoria

Dedico esta tesis principalmente a Dios, por ser mi guía constante y darme la fortaleza necesaria para avanzar en cada etapa de mi vida. A mis padres, Daniel Rengifo Ríos y Ana Lozano Chujutalli, por su amor inquebrantable, su ejemplo de perseverancia y el respaldo firme que siempre me han brindado. A mis hermanos Brayan Rengifo y Hoover Rengifo, por su ánimo y confianza a lo largo del proceso. También, a mi pareja, Greys Cachique por acompañarme en cada momento importante de esta etapa y las etapas que vendrán.

## Agradecimientos

Al Ing. John Clark Santa María Pinedo, por su asesoría constante, sus aportes técnicos y su acompañamiento cercano durante el desarrollo de esta investigación; al Ing. Carlos Mauro Tenazoa, por su colaboración profesional en la validación de las imágenes y su aporte como ingeniero agrónomo en la correcta clasificación del dataset; y al Ing. Dr. Miguel Ángel Vallés Coral, por su guía académica y por contribuir activamente en el proceso de esta tesis, brindándome orientación metodológica y motivación para sostener el enfoque científico a lo largo del trabajo

2

## Índice general

Ficha de identificación.....	6
Dedicatoria .....	7
Agradecimientos .....	8
Índice general.....	9
Índice de tablas .....	11
Índice de figuras.....	12
RESUMEN .....	13
ABSTRACT .....	14
CAPÍTULO I INTRODUCCIÓN A LA INVESTIGACIÓN .....	15
CAPÍTULO II MARCO TEÓRICO .....	19
2.1. Antecedentes de la investigación.....	19
2.2. Fundamentos teóricos.....	21
2.2.1. Presencia de moniliasis en imágenes del fruto de cacao.....	21
2.2.2. Algoritmos basados en extracción de características y regresión para la clasificación de moniliasis .....	21
CAPÍTULO III MATERIALES Y MÉTODOS .....	26
3.1. Ámbito y condiciones de la investigación .....	26
3.1.1. Contexto de la investigación.....	26
3.1.2. Periodo de ejecución .....	26
3.1.3. Autorizaciones y permisos.....	26
3.1.4. Control ambiental y protocolos de bioseguridad.....	26
3.1.5. Aplicación de principios éticos internacionales.....	26
3.2. Sistema de variables .....	27
3.2.1. Variables principales.....	27
3.2.2. Variables secundarias .....	27
3.3. Procedimientos de la investigación.....	27
3.3.1. Diseño de la investigación .....	27

1

44

3.3.2. Actividades del objetivo específico 1: Crear un dataset de imágenes de cacao afectados por moniliasis etiquetadas por profesionales especialistas para entrenar y validar modelos híbridos.....	29
3.3.3. Actividades del objetivo específico 2: Implementar modelos híbridos basado en extracción de características y regresión para la clasificación de moniliasis en imágenes de cacao .....	30
3.3.4. Actividades del objetivo específico 3: Evaluar las métricas de rendimiento de los modelos basados en extracción de características y regresión propuestos sobre la clasificación de imágenes de cacao afectadas por la moniliasis .....	35
<b>CAPÍTULO IV RESULTADOS Y DISCUSIÓN .....</b>	<b>38</b>
4.1. Resultado específico 1: Crear un dataset de imágenes de cacao afectados por moniliasis etiquetadas por profesionales especialistas para entrenar y validar modelos híbridos .....	38
4.2. Resultado específico 2: Implementar modelos híbridos basado en extracción de características y regresión para la clasificación de moniliasis en imágenes de cacao. ....	40
4.3. Resultado específico 3: Evaluar las métricas de rendimiento de los modelos basados en extracción de características y regresión propuestos sobre la clasificación de imágenes de cacao afectadas por la moniliasis .....	49
<b>CONCLUSIONES .....</b>	<b>57</b>
<b>RECOMENDACIONES .....</b>	<b>58</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>59</b>
<b>ANEXOS.....</b>	<b>65</b>
Anexo 1: Autorización para visita de parcelas del ICT .....	66
Anexo 2: Constancia de Revisión de Imágenes.....	67
Anexo 3: Código fuente.....	68

1

## Índice de tablas

Tabla 1 Descripción de variables por objetivo específico .....	27
Tabla 2 Descripción de variables por objetivo específico .....	27
Tabla 3 División del conjunto de evaluación por grupo.....	35
Tabla 4 Vectores de características extraídas .....	46
Tabla 5 Mejores hiperparámetros por regresor .....	47
Tabla 6 Resultados de métricas modelos híbrido .....	48
Tabla 7 Resultados de los modelos híbridos por ciclo métrica accuracy .....	52
Tabla 8 Prueba de homogeneidad de varianza .....	53
Tabla 9 Anova .....	53
Tabla 10 Comparaciones múltiples de Tukey .....	54
Tabla 11 Estadística descriptiva .....	55

52

63

## Índice de figuras

Figura 1: Diagrama de flujo objetivo específico 1 .....	29
Figura 2: Diagrama de flujo objetivo específico 2 .....	31
Figura 3: Extracción de características .....	32
Figura 4: Modelos regresores optimizados .....	33
Figura 5: Validación cruzada k=5 .....	34
Figura 6: Diagrama de Flujo objetivo específico 3 .....	35
Figura 7: Muestra de imágenes de cacao tomados en campo .....	38
Figura 8: Muestra de imágenes de cacao tomados con un fondo sólido .....	38
Figura 9: Revisión de imágenes por especialista.....	39
Figura 10: Cantidad de imágenes por clase.....	41
Figura 11: Cantidad de imágenes por clase despues del aumento de datos.....	42
Figura 12: Segmentación de imágenes.....	43
Figura 13: Pérdida en validación (val_loss) para la mejor configuración .....	45
Figura 14: Curva de aprendizaje del ajuste fino de MobileNetV2.....	45
Figura 15: Gráficos de dispersión Real vs Predicho .....	47
Figura 16: Matrices de confusión.....	48
Figura 17: Distribución de imágenes por clase .....	51
Figura 18: Gráfico de medias.....	55

## RESUMEN

Evaluación de modelos híbridos basado en extracción de características y regresión para clasificación de moniliasis en imágenes de cacao

13

1

La moniliasis, causada por el hongo *Moniliophthora roreri*, es considerada una de las patologías más destructivas que perjudican la producción de cacao en países latinoamericanos, generando pérdidas significativas en la producción. En este contexto, el propósito de este estudio es desarrollar y evaluar modelos híbridos que integren técnicas de extracción de características basadas en redes convolucionales y algoritmos de regresión supervisada para la clasificación de imágenes de frutos de cacao con y sin presencia de moniliasis. La investigación se desarrolló en la región San Martín, Perú, durante el periodo de enero a junio del 2025, utilizando como insumo un dataset conformado por 4000 imágenes divididas en dos clases (sano y monilia), recolectadas tanto en campo como en condiciones controladas, y validadas por un especialista agrónomo. Se aplicó un enfoque metodológico dividido en tres etapas: construcción del dataset, implementación de modelos híbridos y evaluación estadística del rendimiento para determinar el mejor modelo. Las imágenes fueron preprocesadas mediante segmentación y normalización, y posteriormente se extrajeron vectores de características mediante MobileNetV2. Estos vectores fueron utilizados como entrada para los clasificadores SVR, KNN y XGBoost. La validación cruzada por ciclos mostró que el modelo MobileNetV2 + SVR alcanzó la mayor precisión media (98.10%) y la menor desviación estándar (0.0028), demostrando estabilidad durante la replicación. El uso de ANOVA complementando con la prueba de Tukey confirmó la existencia de diferencias significativas entre los modelos evaluados, permitiendo aceptar la hipótesis de investigación. Los resultados indican que el enfoque híbrido propuesto es una solución viable y eficiente para la detección automatizada de moniliasis, con potencial de aplicación en entornos agrícolas reales y posibilidad de adaptación a otras enfermedades o cultivos. Esta investigación contribuye al desarrollo de herramientas inteligentes orientadas al diagnóstico visual en agricultura de precisión, favoreciendo decisiones oportunas y acciones preventivas en la gestión sanitaria del cultivo de cacao.

**Palabras clave:** Mobilenetv2, SVR, KNN Regressor, XGBoost Regressor, Anova.

## ABSTRACT

Evaluation of hybrid models based on feature extraction and regression for the classification of monilia in cocoa images

27 Moniliasis, caused by the fungus *Moniliophthora roreri*, is considered one of the most destructive diseases affecting cocoa production in Latin American countries, resulting in significant yield losses. In this context, the purpose of this study is to develop and evaluate hybrid models that integrate feature extraction techniques based on convolutional networks and supervised regression algorithms for the classification of images of cocoa pods with and without moniliasis. The research was conducted in the San Martín region of Peru from January to June 2025, using a dataset consisting of 4,000 images divided into two classes (healthy and monilia), collected both in the field and under controlled conditions, and validated by an agricultural specialist. A three-stage methodological approach was applied: dataset construction, implementation of hybrid models, and statistical performance evaluation to determine the best model. The images were preprocessed through segmentation and normalization, and feature vectors were subsequently extracted using MobileNetV2. These vectors were used as input for the SVR, KNN, and XGBoost classifiers. Cross-validation by cycles showed that the MobileNetV2 + SVR model achieved the highest mean accuracy (98.10%) and the lowest standard deviation (0.0028), demonstrating stability during replication. The use of ANOVA supplemented with Tukey's test confirmed the existence of significant differences among the evaluated models, allowing the research hypothesis to be accepted. The results indicate that the proposed hybrid approach is a viable and efficient solution for the automated detection of monilia, with potential for application in real-world agricultural settings and the possibility of adaptation to other diseases or crops. This research contributes to the development of intelligent tools for visual diagnosis in precision agriculture, facilitating timely decisions and preventive actions in the disease management of cocoa crops.

40 **Keywords:** Mobilenetv2, SVR, KNN Regressor, XGBoost Regressor, ANOVA.

# CAPÍTULO I

## INTRODUCCIÓN A LA INVESTIGACIÓN

El cacao (*Theobroma cacao L.*), en el mundo, es un cultivo importante, teniendo su origen en América Central y del Sur (Chu et al., 2024). Este cultivo es de gran importancia no solo por su valor económico; sino también, por su relevancia social en las regiones y pueblos productores de cacao (Essah et al., 2022). Por lo cual, en estas áreas, es una fuente principal de empleo e ingresos, teniendo un impacto directo en los hogares de las comunidades respecto en la calidad de vida (Kulesza et al., 2024). Siendo de vital importancia la optimización de estos procesos, haciendo uso de tecnologías que contribuyan a la detección de los problemas que afectan la producción.

La producción de cacao en el año 2023, tuvo un crecimiento del 6.1% respecto al 2022 con un total de 170 000 toneladas (MIDAGRI, 2024). Sin embargo, la calidad y producción de las cosechas de dicho cultivo se vieron perjudicadas por el hongo *Moniliophthora roreri* que provoca la enfermedad de la moniliasis del cacao (Díaz et al., 2022), llegando a causar pérdidas en la producción de hasta el 90% (Plasencia et al., 2022), afectando al cacao en cualquier estado de desarrollo (Flores et al., 2022). Este impacto resalta la necesidad de herramientas eficientes para diagnosticar y clasificar la enfermedad en sus etapas iniciales.

La región con mayor producción es San Martín, siendo el principal productor regional con el 38% de la producción nacional (Diario El Peruano, 2023), pero en el primer semestre del 2023 tuvo una caída de producción con respecto al año 2022 (Dirección General de Políticas Agrarias, 2023). Además, aproximadamente cuenta con 60 mil hectáreas de cacao, donde la moniliasis afecta en promedio el 30% de la producción, que tiende a existir el riesgo que los agricultores dejen de producir cacao al ver que no es factible en términos de ingresos económicos (Leon, 2020), lo que hace evidente la importancia de implementar tecnologías accesibles para apoyar la detección y manejo de la enfermedad.

Los métodos tradicionales para detectar la enfermedad del cacao dependen de diagnósticos visuales realizados por agrónomos, lo que representa un proceso laborioso y propenso a errores (MAMADOU et al., 2023), además la detección temprana de la enfermedad es necesario debido a la rápida propagación de la enfermedad, afectando la producción agrícola y generando preocupación entre los productores (Espinoza-Lozano et al., 2022; Gómez-de la Cruz et al., 2023). Si bien se han logrado avances

tecnológicos en el diagnóstico automatizado de enfermedades agrícolas (Kumi et al., 2022), aunque muchos de estos enfoques no se adaptan a las condiciones del campo.

Para la detección de enfermedades del cacao utilizando el método tradicional, basado en la observación a simple vista, presenta limitaciones en alcance y precisión (Amoako et al., 2023). Por otro lado, el uso de fungicidas químicos para controlar la moniliasis enfrenta desafíos como la resistencia de los patógenos y los impactos ambientales negativos (Valenzuela-Cobos et al., 2023). Esto, sumado al alto costo y al precario conocimiento técnico por parte de los productores (Ccarhuaslla, 2023), subraya la necesidad de soluciones innovadoras basadas en modelos computacionales que sean eficaces y sostenibles.

Los agricultores, en su mayoría, carecen de acceso a herramientas tecnológicas adecuadas para identificar y controlar la moniliasis (Suh & Molua, 2022). Siendo esta situación agravada por la falta de capacitación técnica, lo que limita su capacidad para implementar medidas de control efectivas (Wongnaa et al., 2022). Además, el uso de visión artificial no está aún generalizado, y solo un 20% de los productores utiliza herramientas tecnológicas en procesos de cosecha y postcosecha (Rodríguez-Delzo et al., 2023). Esto impacta la calidad del producto en forma negativa, demostrando la necesidad de optar por soluciones que se puedan adaptar a las condiciones reales del sector agrícola.

La moniliasis genera pérdidas económicas significativas al reducir drásticamente la producción de cacao, afectando especialmente a los pequeños productores que dependen de sus cosechas para subsistir. Además de comprometer los ingresos anuales, esta enfermedad limita la capacidad de inversión en tecnologías preventivas y mejora de cultivos (Reyes et al., 2023). Un manejo adecuado, facilitado por sistemas inteligentes que integren análisis de imágenes y algoritmos predictivos, podría reducir el impacto económico al mejorar el control y la detección temprana de la enfermedad (Anzules-Toala et al., 2021).

La calidad del cacao también se ve afectada por la moniliasis, ya que los frutos dañados no son aptos para el mercado, reduciendo tanto el volumen de producción como el valor del producto final (Delgado-Ospina et al., 2021). Además, puede causar pérdida de semillas de manera total o una disminución en su calidad organoléptica (Soto Chochocca et al., 2022). En este sentido, los modelos basados en extracción y clasificación de imágenes pueden desempeñar un papel fundamental en asegurar la calidad de los productos derivados del cacao.

A pesar de los avances en la clasificación de imágenes mediante modelos híbridos, persiste un vacío en cuanto a la aplicación de estos enfoques en la detección de enfermedades como la moniliasis en cacao en San Martín. Algunas investigaciones han demostrado la eficacia de los modelos híbridos en diferentes áreas, como la clasificación de granos de cacao y otras enfermedades, sin embargo, se basan en datasets generales (Jean et al., 2022; MAMADOU et al., 2023). Aunque los métodos híbridos demuestran eficacia en entornos controlados, su aplicación en el campo se mira limitada por las condiciones variables del área (Hernández Rabadán & Sánchez Hernández, 2022).

Este estudio enfrenta desafíos relacionados con la captura de imágenes en plantaciones de cacao afectadas en San Martín, ya que las condiciones climáticas y de accesibilidad podrían limitar la diversidad del dataset, introduciendo sesgos, mitigados mediante revisiones internas (Adhitya et al., 2020; Mohammad et al., 2022). Además, la implementación del modelo podría enfrentarse a limitaciones prácticas en el campo, como la necesidad de condiciones óptimas para su uso, lo que dificulta su escalabilidad en escenarios reales (Santa-María et al., 2021).

La presente investigación como solución al problema, desarrolló y evaluó modelos híbridos basados en extracción de características y algoritmos de regresión, enfocados en la clasificación de moniliasis en imágenes de cacao, buscando identificar cuál de los modelos híbridos presenta una mejora significativa respecto a los demás (Adhitya et al., 2020; Morales et al., 2024). Esta investigación no contempla una implementación directa en campo, los resultados servirán como base para el desarrollo de soluciones tecnológicas futuras. Por ende, su aplicación dependerá del involucramiento de instituciones interesadas en promover el fortalecimiento de capacidades en el sector agrícola, mediante la capacitación de agricultores. De este modo, se espera contribuir a una detección temprana más efectiva y a una producción sostenible de cacao en la región.

La formulación de la problemática de la investigación fue: ¿Cuál modelo híbrido, basados en extracción de características y técnicas de regresión, tiene mayor precisión en la clasificación de imágenes de frutos de cacao afectadas por la moniliasis?; el cual se trató de responder mediante la hipótesis: existe un modelo híbrido, basado en extracción de características y técnicas de regresión, que presenta una diferencia significativa en la precisión al clasificar imágenes de frutos de cacao afectadas por la moniliasis, en comparación con los demás modelos evaluados.

1 Con respecto a los objetivos, se estableció como objetivo general: Determinar la presencia de moniliasis en imágenes de cacao mediante la evaluación de modelos híbridos basados en extracción de características y regresión. Para cumplir con el primer objetivo específico, se recolectaron imágenes en San Martín, etiquetadas por expertos, generando un dataset equilibrado de 4000 imágenes mediante técnicas de preprocesamiento de datos. El segundo objetivo consistió en ajustar el modelo MobileNetV2 para la extracción de características, integrándolo con modelos de regresión (SVR, KNN, XGBoost). Finalmente, el tercer objetivo implicó evaluar estos modelos utilizando métricas estadísticas, identificando así al modelo más efectivo en la clasificación de moniliasis.

21 A partir del problema identificado y de los objetivos propuestos, este trabajo académico se organiza en cuatro capítulos. El Capítulo I aborda el contexto general y la problemática vinculada a la moniliasis en los frutos de cacao, además de detallar los objetivos específicos que orientaron el desarrollo de la investigación. En el Capítulo II se expone el marco teórico, donde se recopilan antecedentes relevantes y fundamentos conceptuales sobre los modelos híbridos aplicados. El Capítulo III explica con detalle los métodos y procedimientos utilizados para la recolección, tratamiento y evaluación de los datos. Finalmente, el Capítulo IV presenta los principales hallazgos, así como el análisis correspondiente, seguido de las conclusiones, recomendaciones; así como las referencias bibliográficas y anexos correspondientes.

1

2

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. Antecedentes de la investigación

En su investigación (MAMADOU et al., 2023), llevado a cabo en Costa de Marfil, hicieron una evaluación entre la eficacia de utilizar los algoritmos de forma individual y cuando usan un enfoque híbrido, el modelo híbrido está conformado por una CNN siendo MobileNetV2, para extracción de características en imágenes de cacao y SVM, RF, XGBoost, KNN y LR para clasificación que se encarga de indicar a que clase corresponde la imagen. Como dataset para el entrenamiento, validación y pruebas de los modelos utilizaron de kaggle, que contiene 312 imágenes con 1591 objetos clasificados entre tres etiquetas: sanos, monilia y phytophthora. Los resultados muestran que el utilizar un modelo híbrido es eficiente con respecto a un modelo común, siendo específico al utilizar el MobilNetV2 y SVM, consiguió el 86.04% de precisión.

En el estudio (Adhitya et al., 2020), clasificaron imágenes de granos de cacao utilizando un modelo híbrido donde trabajaron con CNN y GLCM para extracción de características; y para la clasificación entre los siete tipos de clases se usó SVM y XGBoost. Llevado a cabo el estudio en Indonesia, donde recolectaron 7428 imágenes, las cuales se dividieron en 75% entrenamiento, 10% pruebas, y un 15% validación. Además, una vez que se obtuvieron las imágenes pasaron por proceso de preprocesamiento, extracción de características, clasificación y el análisis de los resultados. En el resultado se pudo apreciar que al utilizar GLCM y XGBoost se obtuvo una precisión de 65.08% superando al modelo de CNN y SVM que llegó a un 59.14% de precisión, a partir de los resultados obtenidos se visualiza una tasa homogénea con respecto a cada clase.

En la investigación de (Jean et al., 2022), para la clasificación de Granos de Cacao, se evaluó dos modelos híbridos basado extracción (CNN o PCA) y clasificación para granos de cacao (SVM), buscando mejorar la precisión de reconocimiento de calidad de los granos utilizando métodos automáticos, teniendo como lugar de investigación en Costa de Marfil, en donde se recolecto 3470 imágenes de grano de cacao divididas en tres categorías, de las cuales el conjunto de datos se dividió en 60% para entrenamiento, 20% para pruebas y 20% para validación. Los resultados mostraron que el modelo híbrido CNN-SVM con un núcleo RBF y un parámetro  $C=100$  alcanzó la mayor precisión con un 98.32%, superando al modelo híbrido PCA-SVM, destacando la eficacia del

enfoque híbrido para la clasificación, resaltando la viabilidad del uso de este modelo en aplicaciones industriales para mejorar la clasificación automatizada en el procesamiento de cacao.

En su investigación (Mohammad et al., 2022), realizada en Indonesia, se implementó un sistema de detección automática de enfermedades y plagas del cacao mediante extracción de características en espacio HSV y clasificación con el algoritmo k-Nearest Neighbor (KNN). El estudio utilizó 150 imágenes tomadas en campo, clasificadas en tres categorías: podredumbre del fruto, plaga *Conopomorpha cramerella* y chinche chupadora. Tras preprocesar las imágenes, se evaluaron distintos valores de k usando validación Hold-Out y k-fold. El mejor resultado se obtuvo con  $k = 5$ , alcanzando una precisión del 84.44% y hasta 99.33% en validación cruzada, evidenciando la eficacia del uso de KNN y colorimetría para detectar afecciones del cacao.

En la investigación de (KOUASSI et al., 2025), realizada en Costa de Marfil, se diseñó un modelo híbrido para clasificar enfermedades en vainas de cacao, enfocándose en la detección del virus Swollen Shoot. El estudio utilizó 3500 imágenes capturadas en plantaciones de Bouaflé y Divo, correspondientes a las variedades Forastero y Trinitario. El modelo integró DenseNet121 y Vision Transformer como extractores de características, y clasificadores SVM y LightGBM. Las imágenes fueron sometidas a preprocesamiento con CLAHE, normalización y aumento de datos. El mejor desempeño se obtuvo con el enfoque CNN-ViT + SVM-LightGBM, logrando una precisión del 99.53%, F1 Score de 99.62% y un MCC de 0.9901, demostrando alta robustez y aplicabilidad en sistemas inteligentes de diagnóstico agrícola.

En el estudio realizado por (Arboleda et al., 2025) en Filipinas, se propuso un modelo de clasificación de enfermedades del cacao mediante análisis visual multicaracterística y el algoritmo k-Nearest Neighbors (KNN). Se usaron 800 imágenes de Kaggle distribuidas en cuatro clases: sanas, con podredumbre negra, moniliasis y barrenadores de vaina. Las imágenes pasaron por eliminación de fondo y selección de ROI, y se extrajeron características de color (RGB, HSV) y textura (GLCM: contraste, homogeneidad, entropía y energía). Se evaluaron tres variantes de KNN (fine, cosine y weighted), siendo Fine KNN el más preciso con 96.67% en pruebas. El estudio evidenció que la combinación de múltiples atributos visuales con KNN es eficaz para detectar enfermedades del cacao con alta precisión.

8

## 2.2. Fundamentos teóricos

### 2.2.1. Presencia de moniliasis en imágenes del fruto de cacao

#### 2.2.1.1. Moniliasis del Cacao

La moniliasis, es una de las principales enfermedades destructivas que deteriora la producción de cacao en América Latina. Siendo la enfermedad provoca la pudrición de las mazorcas, lo que genera pérdidas significativas en los cultivos, con reducciones de hasta el 80% en algunas regiones. El ciclo de vida del hongo incluye una fase biotrófica en la que invade los tejidos intercelulares de la planta, y una fase necrótica que provoca la destrucción del tejido interno de las mazorcas. Los síntomas pueden tardar hasta 60 días en manifestarse externamente, lo que dificulta su control y manejo eficaz, a esto, sumado las condiciones climáticas, como la alta humedad y las temperaturas cálidas, que conllevan a favorecer el desarrollo de la enfermedad (Espinoza-Lozano et al., 2022).

#### 2.2.1.2. Clasificación de imágenes de cacao

Organizar a las fotografías en categorías como "sano" y "cacao con monilia" implica una serie de pasos esenciales. Este proceso incluye la recolección de datos visuales, seguido del preprocesamiento de las imágenes para normalizarlas y mejorar la calidad visual. A continuación, se realiza la extracción de características específicas, como textura y color, mediante el uso de técnicas como las redes neuronales convolucionales (CNN). Estas características permiten identificar patrones asociados con la presencia de moniliasis, lo que facilita la clasificación. Finalmente, se aplica un algoritmo de clasificación entrenado con etiquetas correspondientes a las dos categorías principales, y se evalúa la precisión del modelo en la detección de mazorcas sanas y afectadas por moniliasis, optimizando así la identificación automática de la enfermedad (MAMADOU et al., 2023).

### 2.2.2. Algoritmos basados en extracción de características y regresión para la clasificación de moniliasis

#### 2.2.2.1. Redes Neuronales Convulsiónales (CNN)

Son modelos de aprendizaje profundo diseñados para el procesamiento de imágenes, y su capacidad para detectar patrones espaciales es esencial en la clasificación de imágenes, las CNN juegan un rol clave al permitir la extracción automática de características visuales complejas, como texturas y anomalías, que facilitan la identificación precisa de la enfermedad. Estas características extraídas se combinarán con técnicas de regresión para mejorar la precisión en la clasificación (Dazzi et al., 2021).

29

### 2.2.2.2. Algoritmos de regresión

Técnicas matemáticas que permiten encontrar la relación entre una variable dependiente continua y una o más variables independientes. Siendo en el aprendizaje automático, utilizado para predecir valores continuos, ajustando un modelo a los datos observados. Los algoritmos conocidos se encuentran la regresión lineal, regresión polinomial y regresión de vectores de soporte (SVR). Estos algoritmos son fundamentales para problemas de predicción, donde se busca identificar tendencias o patrones a partir de un conjunto de datos previamente extraído (Adhitya et al., 2020).

### 2.2.2.3. Modelos híbridos

Los modelos híbridos combinan técnicas, en este caso, extracción de características con algoritmos de regresión para mejorar la precisión en tareas de clasificación, como la identificación de enfermedades en imágenes. En estos modelos, la extracción de características se realiza mediante métodos avanzados como redes neuronales convolucionales (CNN), que permiten identificar patrones visuales complejos a partir de las imágenes. Estas características, que pueden incluir texturas, colores y formas, son transformadas en datos estructurados que sirven como entradas para los algoritmos de regresión (Mohammad et al., 2022).

En el caso de la clasificación, la regresión se utiliza para encontrar la relación entre las características extraídas y las etiquetas de clasificación. Técnicas de regresión como regresión de vectores de soporte (SVR) y XGBoost son comunes en estos enfoques híbridos, ya que permiten ajustar con precisión los datos extraídos a las categorías objetivo, como "sano" o "con monilia" en el contexto de imágenes de cacao. Esta combinación de extracción de características y regresión ha demostrado mejorar significativamente la precisión de los modelos en la clasificación automática de imágenes, al aprovechar las ventajas de ambos métodos (Abdel-salam et al., 2024).

### 2.2.2.4. Métricas de evaluación

#### a) Accuracy

Es una métrica de evaluación común en modelos de clasificación, utilizada para determinar las predicciones correctas que realiza el modelo sobre el total de predicciones realizadas. La fórmula se expresa de la siguiente manera:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Donde:

- **TP (Verdadero Positivo):** Predicciones correctas donde el modelo identifica correctamente las imágenes afectadas por moniliasis como "positivas".

- **TN (Verdadero Negativo):** Predicciones correctas donde el modelo clasifica correctamente las imágenes sanas de cacao como "negativas".
- **FP (Falso Positivo):** Predicciones incorrectas donde el modelo clasifica erróneamente una imagen sana como "positiva".
- **FN (Falso Negativo):** Predicciones incorrectas donde el modelo clasifica erróneamente una imagen afectada como "negativa".

La accuracy es particularmente útil en problemas donde las clases están balanceadas, ya que considera tanto las predicciones correctas para la clase positiva como la clase negativa. Sin embargo, en situaciones donde las clases están desbalanceadas, su uso puede no reflejar adecuadamente el rendimiento del modelo, ya que podría dar una falsa sensación de alto rendimiento al centrarse en la clase mayoritaria (Foody, 2023).

### b) Precision

Evalúa las predicciones positivas correctas realizadas por un algoritmo de clasificación. Se calcula como la proporción de TP sobre el total de instancias clasificadas como positivas, es decir, la suma de TP y FP (Foody, 2023).

La fórmula para calcular la precisión es:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

### c) Recall

Refleja como el algoritmo de clasificación tiene capacidad para poder identificar todas las instancias positivas dentro de un conjunto de datos. Se calcula como la proporción de TP sobre el total de instancias que en realidad son positivas, es decir, la suma de TP y FN (Chicco & Jurman, 2020).

La fórmula es:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

### d) F1 score

Representa el equilibrio entre precision y recall, proporcionando una única métrica que permite evaluar tanto la exactitud como la capacidad del modelo para recuperar instancias positivas. Es especialmente útil cuando se desea un balance entre ambos, particularmente en problemas donde las clases están desbalanceadas y una métrica como la accuracy no proporciona una imagen completa (Chicco & Jurman, 2020). Para mejorar su interpretación en estos casos, se recomienda aplicar técnicas como el reequilibrio del dataset o el uso de funciones de pérdida ponderadas, lo que permite que

el F1 score refleje con mayor fidelidad el rendimiento del modelo ante ambos tipos de clase.

La fórmula es:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Donde:

- **Precision:** Proporción de verdaderos positivos entre todas las predicciones positivas.
- **Recall:** Proporción de verdaderos positivos entre todas las instancias positivas reales.

El F1 score varía entre 0 y 1, donde 1 indica el mejor rendimiento posible, reflejando un balance perfecto entre precisión y recuperación

#### e) MCC

El Matthews Correlation Coefficient (MCC) es una medida utilizada en clasificación binaria que evalúa la calidad del modelo tomando en cuenta las predicciones correctas e incorrectas, tanto para las clases positivas como negativas. El MCC es particularmente efectivo cuando las clases están desbalanceadas, ya que considera los cuatro valores de la matriz de confusión: TP, TN, FP y FN. Para mejorar su eficacia en estos casos, es conveniente aplicar técnicas de reequilibrio de datos que permitan una mejor representación de ambas clases en el entrenamiento.

El MCC se calcula mediante la siguiente fórmula:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

El MCC varía entre -1 y 1, donde 1 indica una predicción perfecta, 0 refleja un modelo que no tiene mejor rendimiento que el azar, y -1 implica una total discordancia entre las predicciones y los datos reales (Chicco & Jurman, 2020).

31

**f) Matriz de confusión**

Constituye una herramienta fundamental para evaluar modelos de clasificación, ya que permite analizar el desempeño del sistema mostrando, en una tabla, las coincidencias y discrepancias entre las clases reales y las clasificadas por el modelo. Esta estructura facilita el cálculo de métricas de evaluación como precisión, recall, F1-score y exactitud. Su formato se basa en una tabla bidimensional, donde las filas indican las clases verdaderas y las columnas corresponden a las clases que el modelo ha predicho (Markoulidakis & Markoulidakis, 2024).

59

	Clase Positiva (Predicha)	Clase Negativa (Predicha)
Clase Positiva (Real)	TP	FN
Clase Negativa (Real)	FP	TN

3

## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1. **Ámbito y condiciones de la investigación**

##### 3.1.1. **Contexto de la investigación**

La investigación se desarrolló en la región San Martín, Perú, en áreas productoras de cacao afectadas por la enfermedad de la monilia. Esta región se caracteriza por un clima tropical, con temperaturas promedio entre 24°C y 28°C, y una alta humedad relativa que varía entre el 75% y el 90%. Estas condiciones climatológicas favorecen el crecimiento del hongo y la propagación de la moniliasis, lo que genera en la producción de cacao, un impacto significativo. Asimismo, la investigación se centra en recolectar imágenes de cacao afectados por la moniliasis para la creación de un dataset, evaluando el rendimiento de modelos híbridos basados en extracción de características y regresión.

##### 3.1.2. **Periodo de ejecución**

La ejecución del proyecto se llevó a cabo durante 6 meses, desde enero del 2025 a junio del 2025.

##### 3.1.3. **Autorizaciones y permisos**

Para la ejecución del proyecto fue necesario contar con la debida autorización de los dueños de las parcelas para poder tomar fotos, de los cuales en el Instituto de Cultivos Tropicales (ICT), se tuvo que presentar un documento donde se dio a conocer el motivo del cual se desea ir a las parcelas, como se muestra en el Anexo 1.

##### 3.1.4. **Control ambiental y protocolos de bioseguridad**

El proyecto no conllevó niveles de riesgo biológico o ambiental, porque tiene un enfoque exclusivo en el análisis y procesamiento de imágenes. Por lo tanto, no fue necesario implementar medidas de manejo ambiental o protocolos de bioseguridad específicos. La investigación se realizó en parte en campo para la recolección de imágenes sin afectar el ambiente, además, la otra parte de la investigación se desarrolla en un entorno digital, sin el uso de sustancias químicas o agentes biológicos que representen riesgo alguno.

##### 3.1.5. **Aplicación de principios éticos internacionales**

Declaro que la intervención de la investigación respetó en todo momento los principios éticos. Particularmente, los principios respeto a las personas, de integridad, respeto al ecosistema, beneficencia y justicia. Además, este proyecto no causó daño alguno, ya

19

2

1

1

57

que se centró en el análisis y procesamiento de imágenes digitales de cacao, sin involucrar la manipulación de seres vivos o el uso de sustancias que afecten al medio ambiente. El estudio se llevó a cabo con total integridad, garantizando la transparencia en los procesos y el respeto a los datos utilizados.

### 3.2. Sistema de variables

#### 3.2.1. Variables principales

**Variable dependiente:** Presencia de moniliasis en imágenes del fruto de cacao.

**Variable independiente:** Algoritmos basados en extracción de características y regresión para la clasificación de moniliasis.

**Tabla 1**

*Descripción de variables por objetivo específico*

**Objetivo específico N° 1:** Crear un dataset de imágenes de cacao afectados por moniliasis etiquetadas por profesionales especialistas para entrenar y validar modelos híbridos.

Variable abstracta	Variable concreta	Medio de registro	Unidad de medida
Presencia de moniliasis en imágenes del fruto de cacao.	Imágenes de frutos de cacao.	Fotografías analizadas por especialistas.	Unidad
	Etiqueta de la imagen	Análisis visual	Categoría Binaria (Cacao sano / Cacao con moniliasis).

**Tabla 2**

*Descripción de variables por objetivo específico*

**Objetivo específico N°2:** Implementar modelos híbridos basado en extracción de características y regresión para la clasificación de moniliasis en imágenes de cacao.

Variable abstracta	Variable concreta	Medio de registro	Unidad de medida
Algoritmos basados en extracción de características y regresión para la clasificación de moniliasis.	Accuracy	Índices estadísticos	Porcentaje
	Precision	Índices estadísticos	Porcentaje
	Recall	Índices estadísticos	Porcentaje
	F1 score	Índices estadísticos	Porcentaje
	MCC	Índices estadísticos	Porcentaje
	Matriz de confusión	Conteo de TP, FP, TN, y FN.	Unidad
	ROC	Umbral de decisión del modelo	Porcentaje

#### 3.2.2. Variables secundarias

No aplica.

### 3.3. Procedimientos de la investigación

#### 3.3.1. Diseño de la investigación

La investigación es de tipo aplicada, siendo adecuada por el uso de conocimientos en inteligencia artificial para la extracción de características y regresión aplicados al dataset

de imágenes de cacao, para clasificar la presencia de Moniliasis con precisión (Oliveira & Silva, 2023).

En cuanto al nivel de la investigación, se clasifica como experimental con un diseño factorial 3x1, porque implica la construcción, prueba y ajuste de modelos híbridos en un entorno controlado, permitiendo manipular la variable independiente y observar su impacto en la precisión de la clasificación de la Moniliasis en las imágenes de cacao, garantizando así la validez y aplicabilidad de los resultados obtenidos en situaciones reales de producción agrícola (Ballance, 2024).

La población está conformada por imágenes de cacao capturadas en la región San Martín, como muestra utilizada para el estudio estará compuesta por 2000 imágenes de cacao, que incluirán tanto plantas sanas como aquellas afectadas por la moniliasis. Posteriormente, estas imágenes fueron sometidas a un preprocesamiento, lo que permitió obtener un buen volumen de datos. Las imágenes están etiquetadas en dos clases: sanos y afectados por moniliasis, lo que garantiza la calidad del dataset y la representatividad de las condiciones reales de los cultivos en la región, considerando imágenes con claridad suficiente para el análisis, excluyendo aquellas con defectos de captura que pudieran interferir con el proceso de clasificación.

El diseño sigue un enfoque metodológico cuantitativo, basado en el análisis experimental factorial 3 x 1, donde se manipuló la variable independiente. Además, se trabajó con los modelos híbridos basados en MobileNetV2 con tres técnicas de regresión diferentes: SVR, KNN, y XGBoost.

La recolección de datos se basará en un muestreo sistemático de imágenes de mazorcas de cacao afectadas por la moniliasis, asegurando una muestra representativa del fenómeno en distintas condiciones climáticas y fases de la enfermedad. Las imágenes se capturaron utilizando una cámara de celular A34 5G Samsung, siguiendo protocolos estrictos de control de calidad y precisión en el etiquetado, realizado por expertos. Para reducir el sesgo de etiquetado, se elaborará una guía con criterios claros (color, textura y patrones característicos) que se aplicará de forma uniforme durante el proceso. (Ayikpa et al., 2023).

En cuanto al diseño experimental, se utilizó validación cruzada para ver el rendimiento de los modelos híbridos. El análisis de los resultados se llevó a cabo mediante la sistematización de las métricas de rendimiento en tablas y gráficos, para luego aplicar un análisis estadístico mediante ANOVA unidireccional, que permitió identificar si existe diferencias significativas entre los modelos lo que permitirá una visualización clara y objetiva de las diferencias entre los modelos (MAMADOU et al., 2023).

### 3.3.2. Actividades del objetivo específico 1: Crear un dataset de imágenes de cacao afectados por moniliasis etiquetadas por profesionales especialistas para entrenar y validar modelos híbridos

Para poder cumplir con el objetivo de crear un dataset de imágenes de cacao de frutos sanos y con monilia, se dividió en 5 etapas: preparación de entorno, captura de imágenes, revisión de imágenes, clasificación de las imágenes por un especialista agronomo y el almacenamiento. Cada uno de las etapas se cumplió para poder garantizar la calidad de los datos que servirá en el desarrollo de los modelos híbridos de extracción de características y regresaaión para la clasificación de monilia en el cacao. Todo el procedimiento en secuencia se muestra en la Figura 1



**Figura 1**

*Diagrama de flujo objetivo específico 1*

#### a) Preparación de entorno

Para iniciar la construcción del conjunto de datos, se identificaron parcelas de cacao en diferentes provincias de la región San Martín, específicamente de las provincias de San Martín, Tocache y Huallaga (para los frutos sanos) y Rioja para los frutos de cacao con moniliasis seleccionadas por presencia de frutos en diversas etapas. Se establecieron criterios técnicos para la toma de imágenes, incluyendo la activación de líneas guía en la cámara del celular y el mantenimiento de una distancia de 15 a 30 cm del fruto. Este enfoque se alinea para mejorar la uniformidad de la captura. Asimismo, resaltan la importancia de definir parámetros de encuadre e iluminación desde la etapa de diseño experimental para garantizar coherencia visual en imágenes agrícolas (Mohammad Yazdi Pusadan et al., 2022).

#### b) Captura de imágenes

Para la captura de imágenes de los frutos de cacao sanos y con monilia se realizó con un celular Samsung Galaxy A34 5G que cuenta con una cámara principal de 48MP, pero

se tomó a 12MP con agrupación de píxeles, garantizando una buena calidad de las imágenes.

La toma de imágenes se realizó en dos escenarios. El primero incluyó la captura directa en campo con iluminación natural, evitando sombras y objetos distractores. En el segundo, los frutos afectados por moniliasis fueron recolectados y fotografiados sobre un fondo gris en ambiente controlado, permitiendo destacar textura y color (Arboleda et al., 2025).

### **c) Revisión de imágenes**

Una vez capturadas, las imágenes fueron sometidas a un proceso de revisión manual con el objetivo de eliminar aquellas con desenfoque, duplicados u otros defectos técnicos. Esta fase de control de calidad fue esencial para asegurar la utilidad de las imágenes en procesos posteriores de análisis (Mohammad Yazdi Pusadan et al., 2022).

### **d) Clasificación de imágenes**

La clasificación de imágenes fue realizada con el apoyo de un ingeniero agrónomo, quien identificó visualmente las condiciones de los frutos y validó la pertenencia de cada imagen a la clase “sano” o “monilia”. Esta participación especializada garantizó etiquetas confiables, requisito crucial para entrenar modelos supervisados (Jean et al., 2022).

### **e) Importación de las imágenes de cacao en Google Drive**

Las imágenes clasificadas fueron organizadas y almacenadas en Google Drive en carpetas separadas por clase, lo cual permitió integrar fácilmente el dataset a plataformas como Google Colab para el entrenamiento de modelos, asegurando la trazabilidad y disponibilidad del conjunto de imágenes (KOUASSI et al., 2025).

### **3.3.3. Actividades del objetivo específico 2: Implementar modelos híbridos basado en extracción de características y regresión para la clasificación de moniliasis en imágenes de cacao**

Para lograr el implementar los modelos híbridos de extracción de características y regresión para clasificar las imágenes, se realizó 3 etapas que abordan el pre-procesamiento de imágenes, procesamiento y reportes. El flujo de este procedimiento se muestra en la

**Figura 2.**

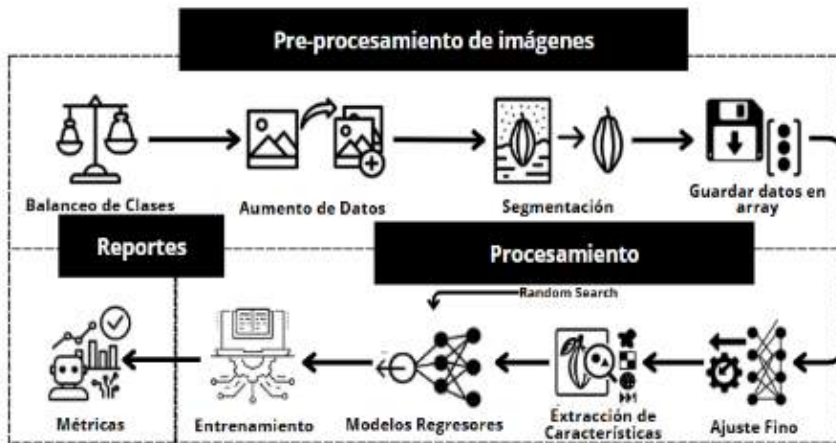
**Figura 2**

Diagrama de flujo objetivo específico 2

### a) Pre-procesamiento de imágenes

- **Balanceo de clases**

Para poder implementar los modelos híbridos es necesario contar con un dataset robusto y de calidad, teniendo en cuenta que la diferencia entre clases sea lo más proporcional posible, por eso que se realiza el balanceo de clases, primero para ver la cantidad de imágenes por clase para luego tomar una decisión con respecto a eso.

- **Aumento de datos**

En esta etapa una vez sabiendo con cuantas imágenes se cuenta por clase se procedió a realizar el aumento de datos que consiste en, apartir de las imágenes originales, realizar transformaciones para que el dataset aumente en cantidad de imágenes, para el proyecto, se utilizó 3 transformaciones que se aplicó de manera aleatoria, entre las cuales está, el volteo en horizontal, rotación con con 25 grados de rotación en sentido horario y un zoom del 15%, al mismo tiempo, todas las imágenes de salida se configuraron para que sean de 224x224 pixeles (MAMADOU et al., 2023).

- **Segmentación para quitar el fondo de las imagenes**

Para la extracción automatizada del fondo de las imágenes de cacao (guardar las imágenes con un fondo transparente), se utilizó la herramienta de eliminación de fondos rembg. Todo el trabajo se hizo conectando con Google Drive. Con el objetivo de que el modelo híbrido se centre en las características del cacao y no en la información de otros objetos (Arboleda et al., 2025).

- **Guardar datos en Array**

Al construir los modelos híbrido de extracción de características y regresión para clasificar la monilia en el cacao es necesario que los datos se guarden en array, donde  $x$  representa a la dimensión de las imágenes y también número de canales de color RGB; en el caso de  $y$  representa a las clases 0 (Sano) y 1 (Monilia).

**b) Procesamiento**

- **Ajuste Fino**

Se vió necesario aplicar ajuste fino para que el modelo extractor pueda reconocer características que tiene las imágenes de cacao capturadas ya que mobilenetv2 es un modelo preentrenado con las imágenes de Imagenet, pero no tiene una clase cacao en su repertorio, entonces para que el modelo sea de calidad es necesario aplicarlo, buscando los mejores hiperparámetros para que al momento de aprender nuevas características tenga pérdida mínima, teniendo los siguientes hiperparámetros:

1. Capas a probar: Para saber cuantas capas de las últimas no se van a congelar para que se ajusten a las imágenes de cacao.
2. Learning rates: Controla qué tan grandes son los pasos que da el optimizador al ajustar los pesos.
3. Droptous: Sirve en regularización para evitar el sobreajuste.

- **Extracción de características**

Se construyó un extractor de características a partir del modelo MobileNetV2 ajustado, eliminando la capa de salida y conservando la penúltima capa densa. Esto permitió obtener vectores numéricos representativos de cada imagen de entrada. Para el extractor se aplicó sobre el conjunto separado para los regresores, generando la matriz  $X\_features$ , donde cada fila corresponde a una imagen y cada columna a una característica aprendida (Jean et al., 2022). Estas características fueron utilizadas como entrada para los modelos de regresión (SVR, KNN, XGBoost), como se muestra en la

**Figura 3.****Figura 3**

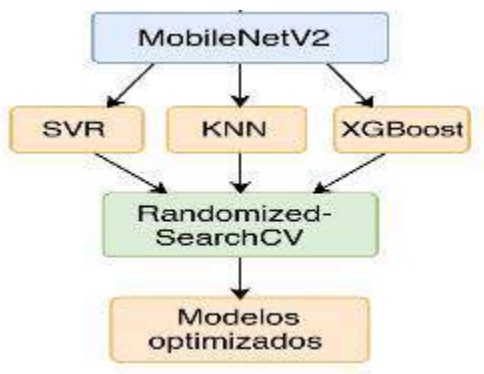
*Extracción de características*

- **Modelos de regresión**

Para la etapa de clasificación se implementaron tres modelos de regresión: Support Vector Regressor (SVR), K-Nearest Neighbors Regressor (KNN) y XGBoost Regressor (XGB). Estos modelos recibieron como entrada los vectores de características generados por MobileNetV2, previamente ajustado con fine-tuning. Con el fin de optimizar el rendimiento de cada modelo, se aplicó la técnica de búsqueda aleatoria de hiperparámetros mediante RandomizedSearchCV, utilizando validación cruzada de cinco pliegues (K=5) y la métrica de error absoluto medio negativo como criterio de evaluación (Adhitya et al., 2020). La búsqueda se realizó sobre los siguientes espacios de hiperparámetros:

- SVR: coeficiente de penalización C, margen de tolerancia epsilon, y parámetro de kernel gamma
- KNN: número de vecinos n\_neighbors, tipo de ponderación weights y métrica de distancia p
- XGBoost: número de árboles n\_estimators, profundidad máxima max\_depth, tasa de aprendizaje learning\_rate y proporción de muestreo subsample

Finalizada la búsqueda, se seleccionó la mejor configuración de cada regresor en función de su rendimiento promedio como se muestra en la Figura 4, siendo estos modelos los que se utilizaron en la etapa de evaluación.

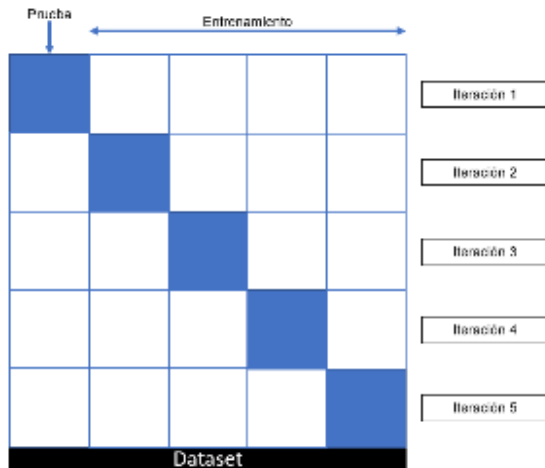


**Figura 4**  
*Modelos regresores optimizados*

- **Entrenamiento**

Identificados los mejores hiperparámetros para cada modelo regresor mediante RandomizedSearchCV, se procedió a evaluar su rendimiento utilizando validación cruzada de cinco pliegues (K=5). Esta técnica consistió en dividir el conjunto de datos en cinco partes iguales, utilizando cada una como conjunto de prueba en una iteración

y las cuatro restantes como conjunto de entrenamiento, como se muestra en la **Figura 5**.



**Figura 5**  
Validación cruzada  $k=5$

### c) Reporte

A partir de las predicciones obtenidas tras la validación cruzada, se realizaron dos tipos de análisis: primero, una evaluación continua del rendimiento de los modelos regresores y luego un análisis binario posterior a la clasificación (Foody, 2023).

#### Análisis previo a la binarización

Antes de convertir las salidas a clases, se analizaron los resultados en su forma continua para evaluar la capacidad de los modelos de aproximarse a la clase objetivo (0 para sano y 1 para moniliasis), utilizando el gráfico de dispersión que muestra la relación entre los valores reales y las predicciones continuas. Una alineación cercana a la diagonal indica buen rendimiento.

#### Transformación a clasificación binaria

Para evaluar la capacidad de los modelos en términos de clasificación, se transformaron las salidas continuas en clases binarias utilizando un umbral de 0.5.

Las métricas consideradas fueron:

- Accuracy: proporción de aciertos globales.
- Precision: exactitud de las predicciones positivas.
- Recall (Sensibilidad): cobertura de los positivos reales.
- F1-Score: equilibrio entre precisión y recall.
- MCC: coeficiente de correlación de Matthews, robusto incluso con clases desbalanceadas.

Matriz de confusión: permite visualizar las verdaderas clases frente a las predicciones binarias, facilitando la interpretación de falsos positivos y falsos negativos.

### 3.3.4. Actividades del objetivo específico 3: Evaluar las métricas de rendimiento de los modelos basados en extracción de características y regresión propuestos sobre la clasificación de imágenes de cacao afectadas por la moniliasis

Para esta etapa, se utilizaron los vectores de características generados previamente con MobileNetV2 durante el proceso de extracción, así como las etiquetas asociadas (0: sano, 1: moniliasis). Estos vectores fueron separados al inicio del experimento y almacenados específicamente para la evaluación final. Se cargaron desde archivos .npy ubicados en Google Drive. El procedimiento seguido para esta evaluación se resume en la **Figura 6**.



**Figura 6**  
Diagrama de Flujo objetivo específico 3

#### a) División del conjunto de evaluación en bloques balanceados

Se utilizó validación cruzada rotativa, donde se dividió el conjunto de evaluación en 17 bloques, cada uno compuesto por muestras balanceadas (sano y monilia). Este procedimiento garantiza que en cada ciclo de evaluación el conjunto de prueba sea diverso. Para ello, se separaron primero las muestras por clase y luego se organizaron en bloques de igual tamaño, como se muestra en la **Tabla 3**.

**Tabla 3**  
División del conjunto de evaluación por grupo

Bloque	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>Grupo 1</b>		n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 2</b>	n		n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 3</b>	n	n		n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 4</b>	n	n	n		n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 5</b>	n	n	n	n		n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 6</b>	n	n	n	n	n		n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 7</b>	n	n	n	n	n	n		n	n	n	n	n	n	n	n	n	n
<b>Grupo 8</b>	n	n	n	n	n	n	n		n	n	n	n	n	n	n	n	n
<b>Grupo 9</b>	n	n	n	n	n	n	n	n		n	n	n	n	n	n	n	n
<b>Grupo 10</b>	n	n	n	n	n	n	n	n	n		n	n	n	n	n	n	n
<b>Grupo 11</b>	n	n	n	n	n	n	n	n	n	n		n	n	n	n	n	n
<b>Grupo 12</b>	n	n	n	n	n	n	n	n	n	n	n		n	n	n	n	n

<b>Grupo 13</b>	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 14</b>	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 15</b>	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
<b>Grupo 16</b>	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

### b) Evaluar validación cruzada por ciclo

Una vez conformados los 17 bloques, se ejecutó una estrategia de validación cruzada por ciclo: en cada una de las 16 iteraciones, se excluyó un bloque distinto, utilizando los 16 restantes como base de entrenamiento y validación.

Sobre el conjunto se aplicó validación cruzada estratificada con  $k=5$ , en cada ciclo se calcularon las siguientes métricas (Foody, 2023):

- Exactitud (Accuracy)
- Precisión
- Recall (Sensibilidad)
- F1 Score
- MCC
- Error cuadrático medio (MSE)
- Área bajo la curva ROC (AUC)

El proceso se automatizó mediante una función que recibe como entrada el modelo entrenado y los bloques  $X_{\text{bloques}}$ ,  $y_{\text{bloques}}$ .

### c) Resultados por modelo

Para cada modelo híbrido (SVR, KNN y XGBoost), se almacenaron los resultados de las 16 corridas en una tabla individual. Cada fila representa un ciclo y contiene las métricas calculadas.

Estas tablas fueron exportadas en formato Excel para facilitar su visualización, análisis estadístico y posterior inclusión como parte del capítulo de resultados. Los DataFrames se estructuraron con columnas como 'Ciclo', 'Accuracy (%)', 'Precision (%)', etc., y cada modelo fue identificado claramente en su respectivo archivo.

### d) Análisis estadístico para selección del mejor modelo

Finalizado el proceso de evaluación, se construyeron tablas con los resultados por ciclo para cada modelo. Estas métricas fueron cargadas al software SPSS para realizar un análisis de varianza (ANOVA de un factor) sobre la métrica Accuracy, debido a que constituye un indicador que permite analizar bajo un conjunto de datos balanceado entre clases, condición en la que la exactitud ofrece una representación válida y robusta del

desempeño global del clasificador (Villalobos-Culqui et al., 2025). Además, se utilizó la prueba HSD de Tukey para identificar diferencias significativas entre pares de modelos. Este análisis permitió determinar qué modelo mostró el mejor desempeño promedio y estabilidad estadística (Huaytalla Oré et al., 2024).

2

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1. Resultado específico 1: Crear un dataset de imágenes de cacao afectados por moniliasis etiquetadas por profesionales especialistas para entrenar y validar modelos híbridos

##### a) Captura de imágenes

- Captura de imágenes de cacao en campo

De la captura de imágenes en campo para los frutos sanos según los criterios aplicados se puede visualizar en la Figura 7.



**Figura 7**

*Muestra de imágenes de cacao tomados en campo*

- Captura de imágenes de cacao con un fondo sólido

Para la captura de imágenes de cacao con fondo, de los frutos de cacao con monilia, según los criterios aplicados se muestra en la Figura 8.



**Figura 8**

*Muestra de imágenes de cacao tomados con un fondo sólido*

##### b) Clasificación de imágenes

c)

Con la ayuda de un ingeniero agrónomo, se procedió a revisar las imágenes en donde se fueron descartando imágenes de frutos que no tenían la enfermedad o no estaban sanos, como se muestra en la Figura 9; además al terminar la revisión se firmó una constancia que acredita la revisión de las imágenes Anexo 2.



**Figura 9**  
*Revisión de imágenes por especialista*

El conjunto obtenido de las tomas de fotografías es de 2184 imágenes, correspondientes a frutos con moniliasis, capturadas sobre fondo gris en condiciones controladas y de frutos sanos tomadas directamente en campo con fondo natural, siendo estas imágenes disponibles en el siguiente link de Google Drive: [https://drive.google.com/drive/folders/1HQy6te4v9g8vwg54tg5jUYVz89gK4JeD?usp=drive\\_link](https://drive.google.com/drive/folders/1HQy6te4v9g8vwg54tg5jUYVz89gK4JeD?usp=drive_link).

La captura en fondo controlado permitió estandarizar la iluminación y minimizar distracciones visuales en las imágenes de monilia, optimizando así la futura extracción de características. Por su parte, el uso de fondo natural para frutos sanos garantizó una representación de condiciones reales de campo, lo cual contribuye a mejorar la capacidad de generalización de los modelos entrenados con este dataset. Además, los criterios seleccionados fortalecen la calidad del dataset, en la investigación de (KOUASSI et al., 2025), destacan la importancia de una recolección visual controlada y sistemática para proyectos de clasificación automatizada en cultivos agrícolas.

A diferencia del estudio realizado por (Mohammad Yazdi Pusadan et al., 2022), en el que se trabajó con un volumen limitado de imágenes (150) y se aplicaron técnicas de colorimetría (HSV), el presente trabajo desarrolló un conjunto más amplio y variado, incorporando diversidad en el entorno de captura y en las condiciones de iluminación. Esto no solo mejoró la representatividad del dataset, sino que también ayudó a tener

menos posibilidad de sobreajuste en los modelos y permitió evaluar su desempeño frente a casos reales. Este enfoque busca alcanzar un balance entre simplicidad técnica, visual y aplicabilidad práctica, para tener una base sólida que servirá el desarrollo de los modelos híbridos orientados a la detección de moniliasis en el cultivo de cacao.

## 4.2. Resultado específico 2: Implementar modelos híbridos basado en extracción de características y regresión para la clasificación de moniliasis en imágenes de cacao.

### a) Pre-procesamiento de imágenes

El preprocesamiento de imágenes fue una etapa fundamental para garantizar la calidad y equilibrio del conjunto de datos antes de aplicar modelos de extracción de características y regresión, comprendiendo esta etapa, el análisis de distribución de clases, aumento de datos y segmentación de fondo.

- **Balanceo de clases**

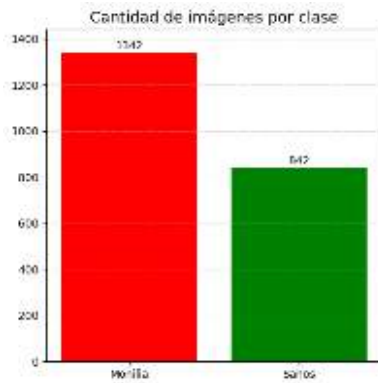
El primer paso consistió en analizar cuántas imágenes se tenían por clase, buscando mantener una distribución equilibrada entre imágenes por clase, esta verificación se realizó mediante una función que contó las imágenes por carpeta y graficó el resultado.

```
clases = ['Monilia', 'Sanos']
cantidades = []
```

Para cada clase, se accede a la carpeta correspondiente, se cuentan las imágenes con extensiones válidas y se guarda el total.

```
for clase in clases:
    ruta_clase = os.path.join(ruta_base, clase)
    imagenes = [f for f in os.listdir(ruta_clase) if
f.lower().endswith(('.jpg', '.jpeg', '.png'))]
    cantidades.append(len(imagenes))
```

Al analizar la distribución original de imágenes por clase como se muestra en la Figura 10, se observó un desequilibrio (1342 imágenes de “Sano” frente a 842 de “Monilia”), pudiendo esta desproporción afectar negativamente el desempeño de los modelos.



**Figura 10**  
*Cantidad de imágenes por clase*

- **Aumento de datos**

Se aumentó el número de imágenes por clase hasta alcanzar 2000 por cada una, aplicando transformaciones geométricas a cada imagen original, como rotaciones, volteo horizontal y zoom, realizando de manera aleatoria por clase.

Se define la carpeta de origen con imágenes originales y la de destino para guardar las aumentadas. Si no existe, se crea automáticamente.

```
carpeta_origen = '/content/drive/MyDrive/Dataset Cacao/Monilia'  
carpeta_destino = '/content/drive/MyDrive/Dataset Cacao Aumento  
Datos/Monilia'  
os.makedirs(carpeta_destino, exist_ok=True)
```

Se fija el número total de imágenes que se desea alcanzar tras el aumento.

```
objetivo_por_clase = 2000
```

Se configura el generador de aumentos con transformaciones geométricas:

```
augmentador = ImageDataGenerator(  
    rotation_range=25,  
    horizontal_flip=True,  
    zoom_range=0.15,  
    fill_mode='nearest'  
)
```

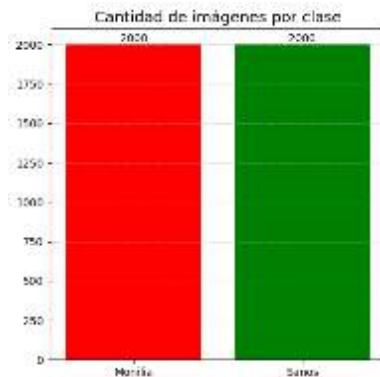
Se lista y cuenta cuántas imágenes originales hay en la carpeta de entrada.

```
imagenes = [f for f in os.listdir(carpeta_origen) if  
f.lower().endswith(('.jpg', '.jpeg', '.png'))]  
num_originales = len(imagenes)
```

Se calcula cuántas veces se debe aumentar cada imagen para alcanzar exactamente las 2000, y cuántas necesitarán un aumento adicional para cubrir el residuo.

```
aumentos_por_imagen = (objetivo_por_clase - num_originales) //  
num_originales  
residuo = (objetivo_por_clase - num_originales) % num_originales
```

Después de aplicar las transformaciones se alcanzó un total de 2000 imágenes por clase como se muestra en la Figura 11, asegurando una distribución balanceada en el dataset final.



**Figura 11**

*Cantidad de imágenes por clase después del aumento de datos*

- **Segmentación de imágenes para quintar fondo**

Con el objetivo de eliminar elementos de fondo irrelevantes y centrar el análisis en el fruto del cacao, se aplicó segmentación utilizando la herramienta rembg.

53

Se definen las carpetas de entrada (imágenes aumentadas) y salida (imágenes segmentadas).

```
carpeta_entrada = '/content/drive/MyDrive/Dataset Cacao Aumento  
Datos/Sanos'  
carpeta_salida = '/content/drive/MyDrive/Dataset Cacao  
Segmentado/Sanos'
```

Se crea una lista con los nombres de todos los archivos de imagen válidos.

18

```
imagenes = [f for f in os.listdir(carpeta_entrada) if  
f.lower().endswith(('.png', '.jpg', '.jpeg'))]
```

Se recorre cada imagen para eliminar el fondo:

39

```
for i, archivo in enumerate(imagenes, start=1):  
    ruta_imagen = os.path.join(carpeta_entrada, archivo)  
    with open(ruta_imagen, 'rb') as f:  
        input_img = f.read()  
        output_img = remove(input_img)
```

La imagen sin fondo es reconvertida a formato RGBA y redimensionada a 224x224 píxeles, lo cual es necesario para que sea compatible con modelos como MobileNetV2.

```
img_sin_fondo = Image.open(io.BytesIO(output_img)).convert("RGBA")
img_sin_fondo = img_sin_fondo.resize((224, 224), Image.LANCZOS)
```

Se asigna un nombre estándar a cada imagen procesada y se guarda en la carpeta correspondiente.

```
nombre_salida = f"sano_{i}.png"
ruta_salida = os.path.join(carpeta_salida, nombre_salida)
img_sin_fondo.save(ruta_salida, format="PNG")
```

Permitiendo aislar los objetos principales dejando el fondo transparente, mejorando la calidad de las características extraídas posteriormente por el modelo MobileNetV2, las imágenes quedan sin fondo como se muestra en la Figura 12.



**Figura 12**

*Segmentación de imágenes*

- **Guardar datos en array**

Se trabaja con las imágenes segmentadas

```
ruta_base = '/content/drive/MyDrive/Dataset Cacao Segmentado'
```

Teniendo en cuenta las dos clases:

```
clases = {'Sanos': 0, 'Monilia': 1}
```

Los datos se van guardando en dos arrays, donde x guarda la dimensión y el número de canales de color RGB; en caso de y representa a las clases 0 (Sano) y 1 (Monilia).

```
X = np.array(X)
y = np.array(y).astype(np.float32)
```

## b) Procesamiento

### • Ajuste fino

Para realizar el ajuste fino se utilizó el 25% del total de dataset

```
X_finetune, X_regresores, y_finetune, y_regresores =  
train_test_split(  
X, y, test_size=0.75, stratify=y, random_state=42)
```

Se tuvo que preparar el dataset para el proceso

```
dataset_ft = tf.data.Dataset.from_tensor_slices((X_finetune,  
y_finetune)).shuffle(len(X_finetune), seed=42)  
train_size = int(0.8 * len(X_finetune))  
train_ds = dataset_ft.take(train_size).batch(32)  
val_ds = dataset_ft.skip(train_size).batch(32)
```

Colocamos hiperparámetros para que encuentre el mejor

```
capas_a_probar = [20, 30, 40]  
learning_rates = [1e-4, 1e-5]  
dropouts = [0.3, 0.5]
```

Utilizamos early stopping para que si en el entrenamiento al pasar las épocas aumenta la pérdida de valor, se detenga el entrenamiento y se guarde la configuración.

```
early_stop = EarlyStopping(monitor='val_loss', patience=3,  
restore_best_weights=True)
```

Se almacenó el modelo para que apartir del ajuste fino extraiga las características el modelo MobileNetV2.

```
mejor_modelo = model
```

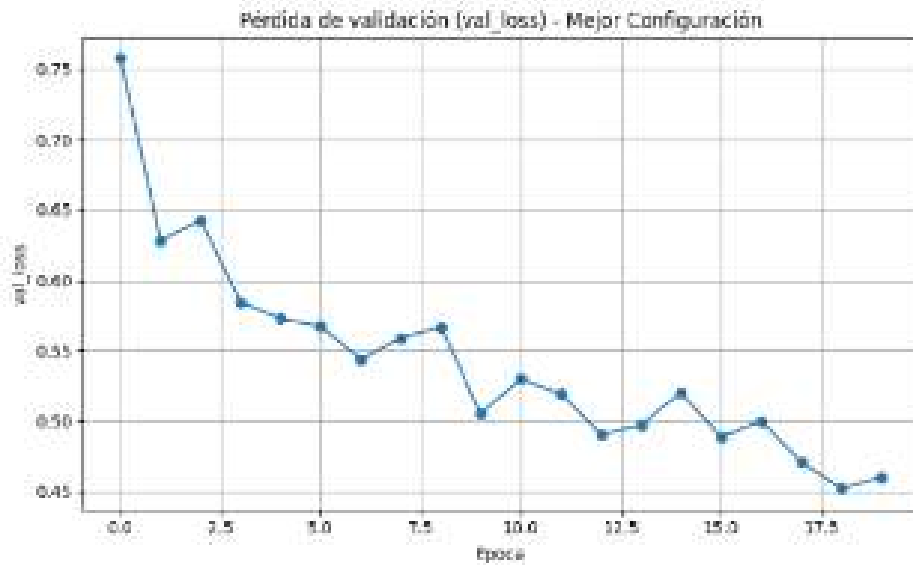
Tras analizar los resultados, se identificó que la mejor configuración correspondió a:

- Capas desbloqueadas: 40
- Learning rate: 1e-05
- Dropout: 0.3

Como se muestra en la Figura 13, la pérdida en validación descendió de forma sostenida hasta estabilizarse en torno a 0.42, reflejando un aprendizaje efectivo y una buena convergencia.

17

25



**Figura 13**  
Pérdida en validación (val\_loss) para la mejor configuración

El seguimiento de estas curvas es esencial para identificar problemas de sobreajuste (overfitting) o subajuste (underfitting), puesto que una divergencia marcada entre ambas indica que el modelo está memorizando los datos de entrenamiento en lugar de generalizar. La Figura 14 muestra la curva de aprendizaje, se observa que ambas curvas siguen una trayectoria decreciente y paralela, sin evidencia de sobreajuste, mostrando que el modelo mantuvo un buen equilibrio durante el entrenamiento.



**Figura 14**  
Curva de aprendizaje del ajuste fino de MobileNetV2

26

- **Extracción de características**

Una vez completado el ajuste fino del modelo MobileNetV2, se extrajeron características tanto para los que serán utilizados para los regresores como también para la evaluación final.

```
X_features_train = feature_model.predict(X_train_regresores,
batch_size=32, verbose=1)
X_features_eval = feature_model.predict(X_eval_final,
batch_size=32, verbose=1)
```

A cada vector se le asoció una etiqueta correspondiente a su clase, codificada como 0 (Sano) o 1 (Monilia), conformando así un conjunto final de 129 columnas por registro: 128 características más 1 columna de clase. Como se observa en la tabla, las características extraídas fueron organizadas en forma tabular, donde cada fila representa una imagen del conjunto de datos.

**Tabla 4**  
*Vectores de características extraídas*

0	1	2	...	125	126	127	label
0	0	15100655	....	0	0	17141268	0.0
0	25045067	0	....	4614421	11193295	0	1.0
0	0	67812586	....	4156477	0	0	0.0
0	0	32732227	....	0	0	0	0.0
0	9621123	0	....	39166865	0	0	1.0
0	27076906	0	....	17854054	8066636	0	1.0

- **Modelos de regresión**

Una vez generados los vectores de características, se procedió a entrenar tres modelos regresores: Support Vector Regressor (SVR), K-Nearest Neighbors Regressor (KNN) y XGBoost Regressor (XGB), mediante una búsqueda aleatoria de hiperparámetros (RandomizedSearchCV) combinada con validación cruzada de cinco pliegues.

```
random_search_svr = RandomizedSearchCV(
    estimator=svr_pipe, param_distributions=param_svr, n_iter=20,
    cv=5, scoring='neg_mean_absolute_error', verbose=2,
    random_state=42,
    n_jobs=-1 )
```

Como se muestra en la Tabla x, se exploraron combinaciones específicas por modelo, seleccionando la que ofreció el menor error absoluto medio (MAE) en validación.

**Tabla 5**  
*Mejores hiperparámetros por regresor*

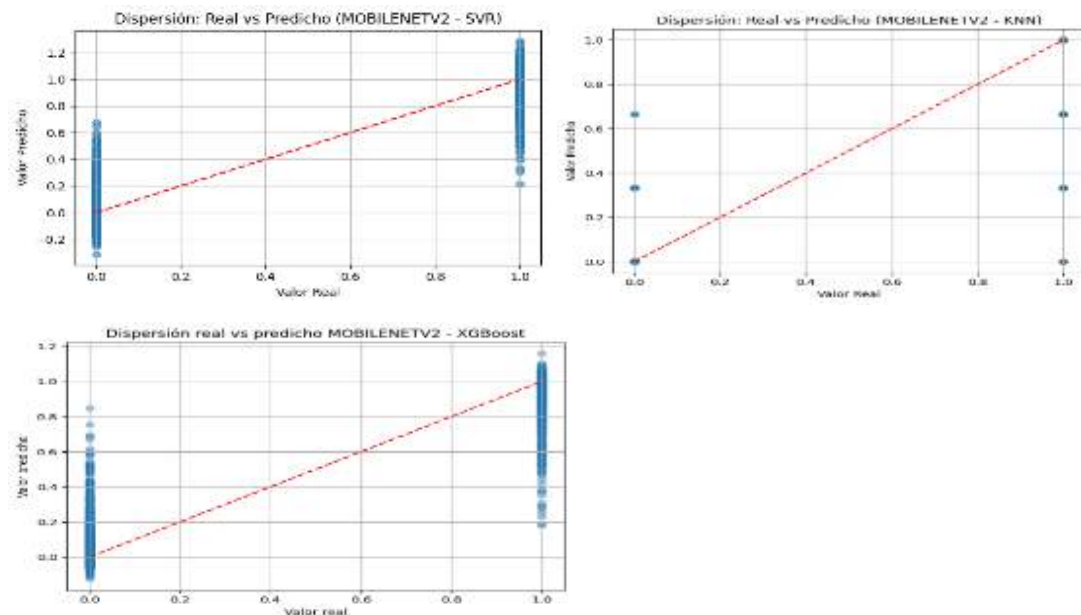
Modelo	Hiperparámetros óptimos
SVR	C = 2.1544, epsilon = 0.0311, gamma = 'scale'
KNN	n_neighbors = 3, weights = 'uniform', p = 1
XGBoost	n_estimators = 200, max_depth = 5, learning_rate = 0.0422, subsample = 0.7

**c) Reportes**

Los resultados que muestran a continuación por cada modelo, fue evaluado en dos etapas: primero como regresor continuo y luego como clasificador binario mediante un umbral de decisión.

- Gráfico de dispersión**

Como se observa en la Figura 15, los gráficos de dispersión muestran una alineación aceptable entre las predicciones del modelo y los valores reales, con mayor concentración alrededor de los extremos 0 y 1, ya que se trata de una clasificación binaria, siendo el modelo híbrido Mobilenetv2 + SVR tienen los datos mejor agrupados por cada clase.

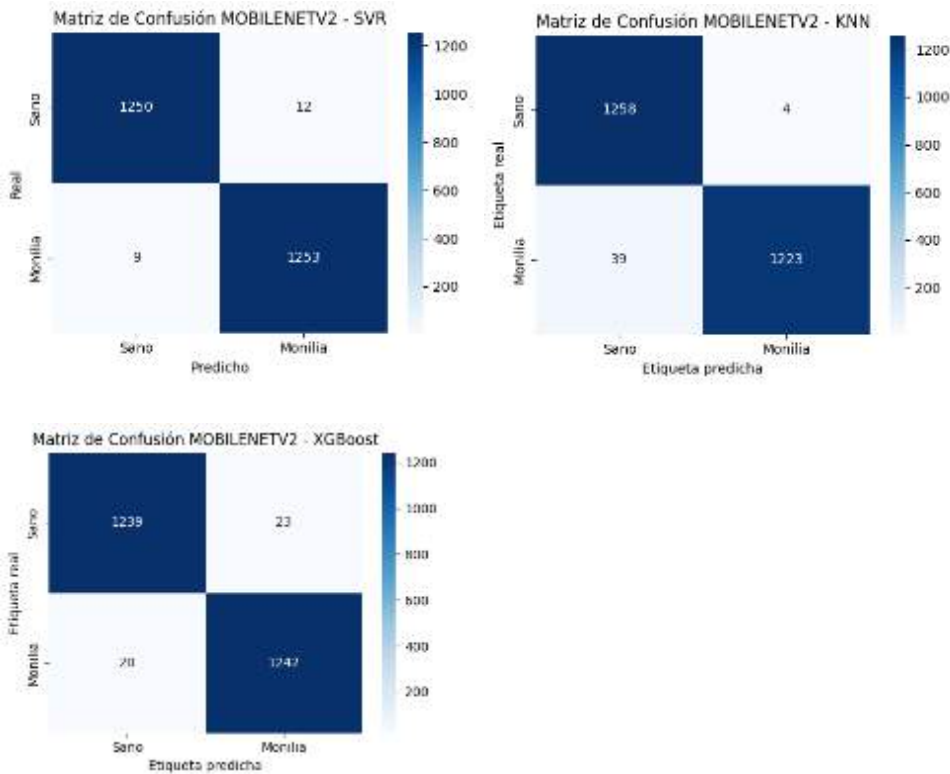


**Figura 15**  
*Gráficos de dispersión Real vs Predicto*

Luego de aplicar el umbral de 0.5, se evaluaron las métricas de clasificación como se muestra en la Tabla x.

• **Matriz de confusión**

La Figura 16 presenta la matriz de confusión, donde se observa que el modelo híbrido Mobilenetv2 + SVR tiene una mejor predicción de las clases.



**Figura 16**  
*Matrices de confusión*

• **Reporte de métricas**

En la Tabla 6 se presentan todas las métricas de evaluación obtenidas para los tres modelos híbridos: MobileNetV2–SVR, MobileNetV2–KNN y MobileNetV2–XGBoost.

**Tabla 6**  
*Resultados de métricas modelos híbrido*

Modelo	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	MCC (%)	MSE
<b>Mobilenetv2 - SVR</b>	99.167987	99.051383	99.286846	99.168975	98.336252	0.025126
<b>Mobilenetv2 - KNN</b>	98.296355	99.674002	96.909667	98.272399	96.629879	0.014879
<b>Mobilenetv2 - XGBoost</b>	98.296355	98.181818	98.415214	98.298378	96.592983	0.021784

La tabla muestra que el modelo híbrido MobileNetV2–SVR es el mejor en todas las métricas, pero es necesario realizar pruebas con la otra parte del dataset para evidenciar si existe una diferencia significativa entre los modelos híbridos.

La aplicación de técnicas de preprocesamiento como el aumento de datos, la segmentación del fondo y la estandarización del tamaño e iluminación de las imágenes resultó fundamental para mejorar la calidad del conjunto de datos, tal como también lo sostienen (Adhitya et al., 2020), quienes resaltan que la transformación geométrica y la limpieza visual previa son clave para optimizar el aprendizaje de modelos en contextos agrícolas.

El uso de MobileNetV2 como extractor de características finamente ajustado permitió obtener representaciones numéricas robustas con 128 atributos por imagen, lo cual coincidió con hallazgos de (MAMADOU et al., 2023), quienes lograron altos niveles de precisión al aplicar ajuste fino y segmentación previa en su modelo CNN para clasificación de enfermedades.

En cuanto al rendimiento de los regresores, el modelo híbrido MobileNetV2–SVR superó a KNN y XGBoost en todas las métricas de clasificación, resultados similares fueron reportados por (MAMADOU et al., 2023), que en su caso emplearon SVM reportando precisiones superiores al 86%.

#### **4.3. Resultado específico 3: Evaluar las métricas de rendimiento de los modelos basados en extracción de características y regresión propuestos sobre la clasificación de imágenes de cacao afectadas por la moniliasis**

##### **a) División del conjunto de evaluación en bloques balanceados**

Para implementar validación cruzada rotativa, se propuso dividir el conjunto de evaluación en 17 bloques balanceados, cada uno con 28 imágenes (14 por clase). Este diseño permite excluir un bloque diferente en cada ciclo de evaluación, garantizando diversidad en los subconjuntos de prueba y equilibrio en la representación de clases.

##### **• Cargar los datos desde Drive**

Se utilizan los archivos .npy previamente generados, que contienen las características extraídas por MobileNetV2 ( $X$ ) y las etiquetas de clase ( $y$ ).

```
X = np.load( "/content/drive/MyDrive/features_eval_moniliasis.npy" )  
y = np.load( "/content/drive/MyDrive/labels_eval_moniliasis.npy" )
```

“ $x$ ” es un arreglo de forma (476, N) que contiene vectores de características numéricas.

“ $y$ ” es un vector binario de longitud 476, donde 0 representa una imagen sana y 1 una imagen con moniliasis.

- **Separar los datos por clase**

Antes de realizar la división en bloques, es necesario separar las muestras por clase para garantizar balance en cada subconjunto.

```
X_sano = X[y == 0]
X_monilia = X[y == 1]
y_sano = y[y == 0]
y_monilia = y[y == 1]
```

Con esta separación, se crean dos subconjuntos:

X\_sano, y\_sano: contienen 238 muestras de la clase sana.

X\_monilia, y\_monilia: contienen 238 muestras de la clase moniliasis.

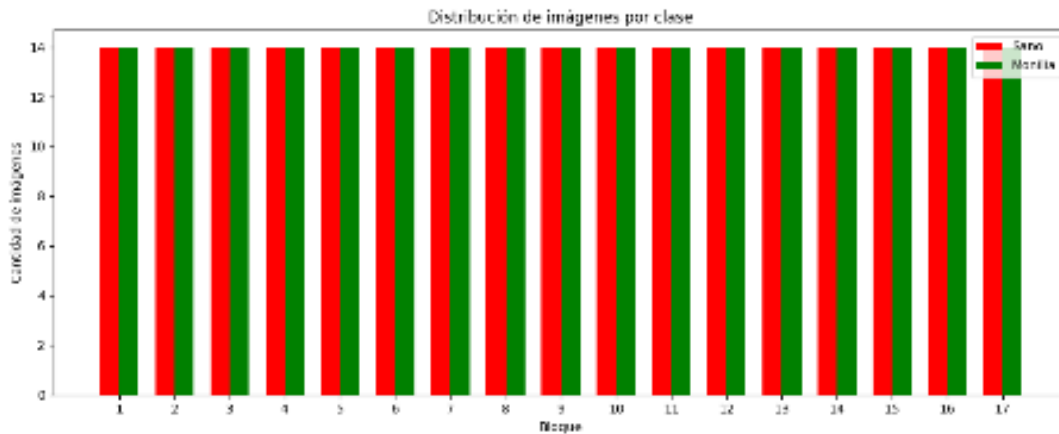
- **Dividir en 17 bloques balanceados**

Se utiliza un bucle for para construir 17 bloques, donde en cada uno se combinan 14 imágenes sanas y 14 con moniliasis.

```
X_bloques, y_bloques = [], []
for i in range(bloques):
    X_b = np.concatenate([X_sano[i*14:(i+1)*14],
X_monilia[i*14:(i+1)*14]])
    y_b = np.concatenate([y_sano[i*14:(i+1)*14],
y_monilia[i*14:(i+1)*14]])
    X_bloques.append(X_b)
    y_bloques.append(y_b)
```

- En cada iteración se toma una porción de 14 elementos por clase.
- Se concatenan para formar un bloque de 28 muestras.
- Luego se añaden a las listas X\_bloques y y\_bloques.

Este proceso se repite 17 veces para cubrir la totalidad del conjunto, en la Figura 17 se muestra la cantidad imágenes extraídas sus características por cada clase.



**Figura 17**

*Distribución de imágenes por clase*

## b) Evaluar validación cruzada por ciclo

Una vez generados los 17 bloques balanceados en el apartado anterior, se procedió a implementar una estrategia de validación cruzada por ciclo. Esta técnica consiste en ejecutar 16 iteraciones, donde en cada una se excluye un bloque distinto como conjunto de descarte, y los 16 bloques restantes son utilizados como base para evaluar el rendimiento de los modelos mediante validación cruzada estratificada con  $k=5$ .

- **Evaluación por ciclo**

Para automatizar este procedimiento, se definió una función que recibe como entrada un modelo previamente entrenado y los bloques  $X_{\text{bloques}}$  y  $y_{\text{bloques}}$ . En cada ciclo, se excluye un bloque diferente y se entrena el modelo usando los otros 16 mediante validación cruzada con cinco particiones.

```
def evaluar_por_bloque(modelo, X_bloques, y_bloques):
    resultados = []
    for i in range(16):
        X_eval = np.concatenate([X_bloques[j] for j in range(17) if
j != i])
        y_eval = np.concatenate([y_bloques[j] for j in range(17) if
j != i])
```

El modelo es evaluado sobre este conjunto utilizando validación cruzada con 5 particiones. Se generan predicciones continuas y luego se convierten a etiquetas binarias usando un umbral de 0.5.

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)
y_pred = cross_val_predict(modelo, X_eval, y_eval, cv=kf,
n_jobs=-1)
y_pred_bin = (y_pred >= 0.5).astype(int)
```

En cada ciclo se calculan métricas de clasificación: exactitud (accuracy), precisión, sensibilidad (recall), F1 score, coeficiente MCC, error cuadrático medio (MSE) y área bajo la curva ROC (AUC).

### c) Resultados por modelo

Se presentan los resultados obtenidos tras aplicar la estrategia de validación cruzada por ciclo, descrita en el apartado anterior. Para cada modelo híbrido se realizaron 16 ciclos de evaluación, utilizando siempre 16 de los 17 bloques como base de validación cruzada con  $k=5$ . La Tabla 7 muestra las métricas de clasificación obtenidas de los modelos híbridos, cada uno de los 16 ciclos de evaluación, se presentan los valores de Accuracy, todos calculados sobre la predicción binarizada a partir de la salida regresiva del modelo.

**Tabla 7**

*Resultados de los modelos híbridos por ciclo métrica accuracy*

Ciclo	Mobilenetv2 + SVR	Mobilenetv2 + KNN	Mobilenetv2 + XGBoost
1	0,9799	0,9888	0,9688
2	0,9799	0,9777	0,9665
3	0,9777	0,9777	0,9576
4	0,9844	0,9754	0,9643
5	0,9821	0,9844	0,9598
6	0,9777	0,9777	0,9643
7	0,9821	0,9821	0,9665
8	0,9844	0,9799	0,971
9	0,9777	0,9821	0,9598
10	0,9799	0,9732	0,9531
11	0,9799	0,9754	0,9598
12	0,9777	0,9866	0,971
13	0,9799	0,9799	0,9732
14	0,9844	0,9844	0,9688
15	0,9821	0,9821	0,9665
16	0,9866	0,9844	0,9665

### d) Análisis estadístico para selección del mejor modelo

Con el fin de verificar si las diferencias en los valores de exactitud (Accuracy) entre los modelos evaluados eran significativas desde el punto de vista estadístico, se utilizó una prueba de análisis de varianza (ANOVA), complementada con un análisis post hoc empleando el método HSD propuesto por Tukey para comparaciones múltiples.

#### Hipotesis

Existe un modelo híbrido, basado en extracción de características y técnicas de regresión, que presenta una diferencia significativa en la precisión al clasificar imágenes

de frutos de cacao afectadas por la moniliasis, en comparación con los demás modelos evaluados.

### Prueba de homogeneidad de varianza

Previo a la aplicación del análisis de varianza (ANOVA), se evaluó si los tres modelos híbridos (MobileNetV2 + SVR, KNN y XGBoost) presentan varianzas homogéneas en cuanto a sus niveles de precisión (accuracy). Para este fin, se aplicó la prueba de Levene bajo diferentes criterios (media, mediana, media recortada), con el objetivo de verificar el cumplimiento del supuesto de homogeneidad de varianzas, requerido para la validez del ANOVA.

**Tabla 8**  
*Prueba de homogeneidad de varianza*

	Estadístico de Levene	gl1	gl2	Sig.	
Accuracy	Se basa en la media	3,004	2	45	,060
	Se basa en la mediana	2,071	2	45	,138
	Se basa en la mediana y con gl ajustado	2,071	2	33,374	,142
	Se basa en la media recortada	2,929	2	45	,064

Dado que los valores de significancia (Sig.) en todos los criterios son mayores a 0.05, se concluye que no existen diferencias significativas entre las varianzas de los tres grupos comparados, en consecuencia, se rechaza la hipótesis alterna (que plantea desigualdad de varianzas) y se acepta la hipótesis nula, la cual afirma que las varianzas son homogéneas, permitiendo proceder con el análisis ANOVA, para poder validar este supuesto es fundamental en estudios comparativos de rendimiento, como lo destacan (Arboleda et al., 2025) quienes verificaron la homogeneidad de varianza antes de comparar la precisión de clasificadores aplicados a imágenes de frutos de cacao en distintas condiciones visuales.

### Anova

Después de confirmar que las varianzas eran homogéneas, se procedió a realizar un análisis de varianza unidireccional (ANOVA), con el objetivo de identificar si las diferencias observadas en la precisión de los tres modelos híbridos eran estadísticamente relevantes.

**Tabla 9**  
*Anova*

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
<b>Entre grupos</b>	27,465	2	13,732	71,776	,000
<b>Dentro de grupos</b>	8,609	4	,191		
<b>Total</b>	36,074	5			
		4			
		7			

El resultado de significancia  $< 0.05$  evidencia la presencia de diferencias estadísticamente relevantes entre al menos dos de los modelos evaluados. En consecuencia, se descarta la hipótesis nula, la cual asume que todos los modelos presentan un desempeño equivalente, y se respalda la hipótesis alternativa que indica que al menos uno de ellos muestra una precisión significativamente distinta respecto a los demás.

### Comparaciones múltiples de Tukey

Tras confirmar la existencia de diferencias estadísticamente relevantes mediante el análisis ANOVA, se procedió a utilizar la prueba de comparaciones múltiples de Tukey con el fin de determinar entre qué pares de modelos híbridos se evidencian discrepancias significativas en su rendimiento.

**Tabla 10**  
*Comparaciones múltiples de Tukey*

(I) Modelos	(J) Modelos	Diferencia de medias (I-J)	Desv . Error	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
<b>Mobilen etv2 - SVR</b>	Mobilenetv2 - KNN	,0002875	,0015479	,981	-,003464	,004039
	Mobilenetv2 - XGBOOST	,0161813*	,0015479	,000	,012430	,019933
<b>Mobilen etv2 - KNN</b>	Mobilenetv2 - SVR	-,0002875	,0015479	,981	-,004039	,003464
	Mobilenetv2 - XGBOOST	,0158938*	,0015479	,000	,012142	,019645
<b>Mobilen etv2 - XGBOOST</b>	Mobilenetv2 - SVR	-,0161813*	,0015479	,000	-,019933	-,012430
	Mobilenetv2 - KNN	-,0158938*	,0015479	,000	-,019645	-,012142

El análisis post hoc confirma que entre los modelos de Mobilenetv2 – SVR y Mobilenetv2 – KNN no tienen diferencia significativa, entonces, para saber cual de los dos modelos es mejor es necesario utilizar estadística descriptiva para hallar cual modelo es mejor.

### Estadística descriptiva

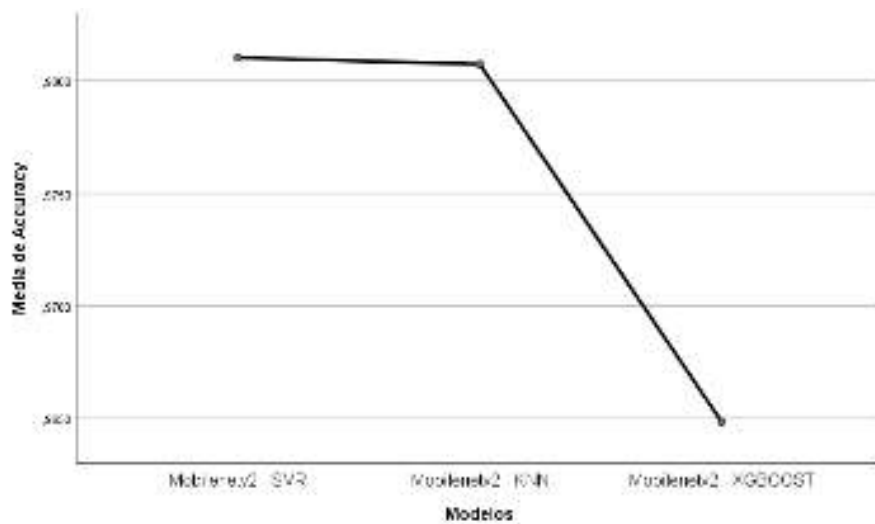
Seguidamente, se muestran los valores descriptivos obtenidos para los tres modelos híbridos analizados, considerando medidas como el promedio, la desviación típica y los intervalos de confianza, con el propósito de examinar su nivel de consistencia y variabilidad a lo largo de las distintas repeticiones realizadas.

41

**Tabla 11**  
*Estadística descriptiva*

	N	Me dia	Desv. Desvia ción	Desv. Error	95% del intervalo de confianza para la media		Míni mo	Máxi mo
					Lími te infer ior	Límit e super ior		
<b>Mobilenetv2 - SVR</b>	1 6	,98 102 5	,00281 98	,00070 50	,979 522	,9825 28	,9777	,986 6
<b>Mobilenetv2 - KNN</b>	1 6	,98 073 8	,00438 22	,00109 56	,978 402	,9830 73	,9732	,988 8
<b>Mobilenetv2 - XGBOOST</b>	1 6	,96 484 4	,00550 88	,00137 72	,961 908	,9677 79	,9531	,973 2
<b>Total</b>	4 8	,97 553 5	,00876 00	,00126 44	,972 992	,9780 79	,9531	,988 8

Según la desviación estándar, el modelo MobileNetV2 + SVR presenta la menor dispersión en sus resultados (0.0028), indicando que es más estable y consistente en las repeticiones, le siguen KNN (0.0044) y XGBoost (0.0055), que mostraron mayor variabilidad, siendo corroborado por el gráfico de medias como se muestra en la Figura 18. Además, el IC: [0.979 – 0.9825], muestra una alta precisión y poca variabilidad en sus resultados.



**Figura 18**  
*Gráfico de medias*

Según el análisis estadístico realizado, se concluye que el modelo híbrido Mobilenetv2 + SVR es el mejor modelo híbrido para la detección de moniliasis en imágenes de cacao,

presentando la mayor media de precisión (98.10%), una desviación mínima y un desempeño estable a lo largo de los 16 ciclos de validación, respaldado por la investigación de (MAMADOU et al., 2023), el enfoque híbrido alcanzó un 86.04% de precisión, destacando entre la extracción de características profunda y los algoritmos de clasificación tradicional.

## CONCLUSIONES

La investigación logra conformar un dataset de 4000 imágenes de frutos de cacao de la variedad CCN – 51, capturadas bajo condiciones naturales y controladas, y validadas por un especialista agrónomo. Esta colección de imágenes, estructurada en las clases “sano” y “monilia”, representa una base sólida y representativa para tareas de entrenamiento, evaluación y aplicación de modelos de clasificación de enfermedades agrícolas.

A través de la combinación de MobileNetV2 con regresores para clasificar, el enfoque híbrido desarrollado demuestra eficiencia en el procesamiento de imágenes agrícolas, integrando extracción de características profunda con algoritmos de regresión como SVR, KNN y XGBoost. Esta estrategia permite reducir la complejidad computacional manteniendo altos niveles de precisión en tareas de clasificación.

Destacando entre los modelos evaluados, MobileNetV2 + SVR alcanza un rendimiento sobresaliente con una precisión media del 98.10% durante la validación cruzada por ciclos, manteniendo además la menor desviación estándar (0.0028). Esta combinación evidencia una alta estabilidad estadística y un desempeño robusto frente a variaciones del conjunto de datos, lo cual se respalda en su IC: [0.979 – 0.9825], indicando que sus resultados son consistentes y presentan baja incertidumbre.

Los resultados obtenidos permiten aceptar la hipótesis planteada, al demostrarse mediante ANOVA ( $F = 71.77$ ;  $\text{Sig.} = 0.000$ ) y prueba de Tukey que el modelo MobileNetV2 + SVR presenta diferencias estadísticamente significativas respecto a los demás modelos. Esta diferencia sustancial justifica su elección como la opción más adecuada dentro del enfoque híbrido propuesto.

## RECOMENDACIONES

Ampliar el dataset con imágenes de distintas variedades de cacao y en diferentes etapas del fruto y la enfermedad, para mejorar la capacidad de generalización del modelo y permitir aplicaciones más robustas en sistemas reales de monitoreo agrícola.

Desarrollar una aplicación móvil que permita a agrícolas capturar imágenes y obtener una predicción inmediata sobre la presencia de moniliasis, promoviendo así la transferencia tecnológica y el uso práctico del modelo propuesto.

Realizar estudios comparativos entre modelos híbridos y modelos end-to-end completamente profundos, como EfficientNet o ResNet50 entrenados desde cero, para evaluar el costo computacional frente al rendimiento en precisión y generalización.

Replicar el análisis estadístico en conjuntos de datos externos o independientes, aplicando nuevamente pruebas como ANOVA y Tukey, con el fin de validar la solidez del modelo MobileNetV2 + SVR en contextos distintos y verificar si mantiene diferencias significativas frente a otras configuraciones, lo cual afianzaría su aplicabilidad en entornos reales.

## REFERENCIAS BIBLIOGRÁFICAS

- Abdel-salam, M., Kumar, N., & Mahajan, S. (2024). A proposed framework for crop yield prediction using hybrid feature selection approach and optimized machine learning. *Neural Computing and Applications*, 36(33), 20723–20750. <https://doi.org/10.1007/s00521-024-10226-x>
- Adhitya, Y., Prakosa, S. W., Köppen, M., & Leu, J.-S. (2020). Feature Extraction for Cocoa Bean Digital Image Classification Prediction for Smart Farming Application. *Agronomy*, 10(11), 1642. <https://doi.org/10.3390/agronomy10111642>
- Amoako, P. Y. O., Cao, G., & Arthur, J. K. (2023). An Image-Based Cocoa Diseases Classification Based on an Improved Vgg19 Model. In *Sustainable Education and Development – Sustainable Industrialization and Innovation* (pp. 711–722). Springer International Publishing. [https://doi.org/10.1007/978-3-031-25998-2\\_55](https://doi.org/10.1007/978-3-031-25998-2_55)
- Anzules-Toala, V., Pazmiño-Bonilla, E., Alvarado-Huamán, L., Borjas-Ventura, R., Castro-Cepero, V., & Julca-Otiniano, A. (2021). Control of cacao (*Theobroma cacao*) diseases in Santo Domingo de los Tsachilas, Ecuador. *Agronomía Mesoamericana*, 45939. <https://doi.org/10.15517/am.v33i1.45939>
- Arboleda, E., Rodrin, S. G., & San Diego, E. C. (2025). Classification of Prominent Cacao Pod Diseases Using Multi-Feature Visual Analysis and k-Nearest Neighbors Algorithm. *ITEGAM- Journal of Engineering and Technology for Industrial Applications* (ITEGAM-JETIA, 11(51). <https://doi.org/10.5935/jetia.v11i51.1367>
- Ayikpa, K. J., Mamadou, D., Gouton, P., & Adou, K. J. (2023). Classification of Cocoa Pod Maturity Using Similarity Tools on an Image Database: Comparison of Feature Extractors and Color Spaces. *Data*, 8(6), 99. <https://doi.org/10.3390/data8060099>
- Ballance, O. J. (2024). Sampling and randomisation in experimental and quasi-experimental CALL studies: Issues and recommendations for design, reporting, review, and interpretation. *ReCALL*, 36(1), 58–71. <https://doi.org/10.1017/S0958344023000162>

- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 6. <https://doi.org/10.1186/s12864-019-6413-7>
- Chu, N. T. B., Le, M. T., La, H. V., Le, Q. T. N., Le, T. D., Tran, H. T. T., Tran, L. T. M., Le, C. T., Nguyen, D. V., Cao, P. B., & Chu, H. D. (2024). Genome-wide identification, characterization, and expression analysis of the small auxin-up RNA gene family during zygotic and somatic embryo maturation of the cacao tree (*Theobroma cacao*). *Genomics & Informatics*, 22(1), 2. <https://doi.org/10.1186/s44342-024-00003-6>
- Dazzi, M., Sebastian, A., Benini, L., & Eleftheriou, E. (2021). Accelerating Inference of Convolutional Neural Networks Using In-memory Computing. *Frontiers in Computational Neuroscience*, 15. <https://doi.org/10.3389/fncom.2021.674154>
- Delgado-Ospina, J., Molina-Hernández, J. B., Chaves-López, C., Romanazzi, G., & Paparella, A. (2021). The Role of Fungi in the Cocoa Production Chain and the Challenge of Climate Change. *Journal of Fungi*, 7(3), 202. <https://doi.org/10.3390/jof7030202>
- Diario El Peruano. (2023). *Midagri está comprometido en impulsar la competitividad de la cadena del valor del cacao*. Economía.
- Díaz, J., Zambrano, R., Cedeño-Amador, S., Córdova-Bermejo, U., Casas, G. G., García-Zurita, N., Sánchez-Arévalo, J. A. J., Arévalo-Gardini, E., Dávila, D., Ruiz, J., Pinchi-Dávila, X., Quispe-Chacón, Z. R., Chia-Wong, J. A., Hurtado-Gonzales, O. P., Rodríguez-Callañaupa, C. A., Maldonado-Fuentes, C., Pérez-Callizaya, E., Leiva-Espinoza, S., Oliva-Cruz, M., ... Aime, M. C. (2022). Diversity in the invasive cacao pathogen *Moniliophthora roreri* is shaped by agriculture. *Plant Pathology*, 71(8), 1721–1734. <https://doi.org/10.1111/ppa.13603>
- Dirección General de Políticas Agrarias. (2023). *Observatorio de Commodities*. Ministerio de Desarrollo Agrario y Riego.
- Espinoza-Lozano, F., Amaya-Márquez, D., Pinto, C. M., Villavicencio-Vásquez, M., Sosa del Castillo, D., & Pérez-Martínez, S. (2022). Multiple Introductions of *Moniliophthora roreri* from the Amazon to the Pacific Region in Ecuador and Shared High Azoxystrobin Sensitivity. *Agronomy*, 12(5), 1119. <https://doi.org/10.3390/agronomy12051119>

- Essah, R., Anand, D., & Singh, S. (2022). An intelligent cocoa quality testing framework based on deep learning techniques. *Measurement: Sensors*, *24*, 100466. <https://doi.org/10.1016/j.measen.2022.100466>
- Flores, V., Gómez Rodríguez, L., López García, J. A., & Grajales Conesa, J. (2022). Mecanismos de infección endógena en frutos de cacao con *Moniliophthora roreri*. *Polibotánica*, *0*(53). <https://doi.org/10.18387/polibotanica.53.13>
- Foody, G. M. (2023). Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient. *PLOS ONE*, *18*(10), e0291908. <https://doi.org/10.1371/journal.pone.0291908>
- Gómez-de la Cruz, I., Chávez-Ramírez, B., Avendaño-Arrazate, C. H., Morales-García, Y. E., Muñoz-Rojas, J., & Estrada-de los Santos, P. (2023). Optimization of *Paenibacillus* sp. NMA1017 Application as a Biocontrol Agent for *Phytophthora tropicalis* and *Moniliophthora roreri* in Cacao-Growing Fields in Chiapas, Mexico. *Plants*, *12*(12), 2336. <https://doi.org/10.3390/plants12122336>
- Hernández Rabadán, D. L., & Sánchez Hernández, J. P. (2022). Redes neuronales convolucionales para el reconocimiento de imágenes con presencia de cenicilla polvorienta en cultivos de tomate. *Programación Matemática y Software*, *14*(3), 21–28. <https://doi.org/10.30973/progmat/2022.14.3/3>
- Huaytalla Oré, L. F., Matta Morales, C., & Delgado-Cotrino, L. (2024). Precisión en la posición de análogos de pilares cortos en el modelo de trabajo ferulizados con hilo dental versus barras de Bis-acryl en mandíbulas edéntulas totales. *Revista Estomatológica Herediana*, *34*(2), 117–125. <https://doi.org/10.20453/reh.v34i2.5528>
- Jean, A. K., Diarra, M., Bakary, B. A., Pierre, G., & Jérôme, A. K. (2022). Application based on Hybrid CNN-SVM and PCA-SVM Approaches for Classification of Cocoa Beans. *International Journal of Advanced Computer Science and Applications*, *13*(9). <https://doi.org/10.14569/IJACSA.2022.0130927>
- KOUASSI, K. S., DIARRA, M., EDI, K. H., & Jean-Claude, K. B. (2025). Detection of cocoa pod diseases using a hybrid feature extractor combining CNN and vision transformer with dual classifier. *Edelweiss Applied Science and Technology*, *9*(1), 668–681. <https://doi.org/10.55214/25768484.v9i1.4209>
- Kulesza, E., Thomas, P., Prewitt, S. F., Shalit-Kaneh, A., Wafula, E., Knollenberg, B., Winters, N., Esteban, E., Pasha, A., Provart, N., Praul, C., Landherr, L.,

- dePamphilis, C., Maximova, S. N., & Gultinan, M. J. (2024). The cacao gene atlas: a transcriptome developmental atlas reveals highly tissue-specific and dynamically-regulated gene networks in *Theobroma cacao* L. *BMC Plant Biology*, 24(1), 601. <https://doi.org/10.1186/s12870-024-05171-9>
- Kumi, S., Kelly, D., Woodstuff, J., Lomotey, R. K., Orji, R., & Deters, R. (2022). Cocoa Companion: Deep Learning-Based Smartphone Application for Cocoa Disease Detection. *Procedia Computer Science*, 203, 87–94. <https://doi.org/10.1016/j.procs.2022.07.013>
- Leon, J. (2020). *San Martín: plagas como el mazorquero y la moniliasis afectaron un 30% de la producción de cacao en el Huallaga Central*. Agencia Agraria de Noticias.
- MAMADOU, D., AYIKPA, K. J., BALLO, A. B., & KOUASSI, B. M. (2023). Cocoa Pods Diseases Detection by MobileNet Confluence and Classification Algorithms. *International Journal of Advanced Computer Science and Applications*, 14(9). <https://doi.org/10.14569/IJACSA.2023.0140937>
- María Aurolina Ccarhuaslla Ochoa. (2023). *Control de enfermedades del cacao (Theobroma cacao L.) en la provincia de Mariscal de Cáceres, región San Martín año 2022* [Universidad Nacional de San Martín]. <https://repositorio.unsm.edu.pe/bitstream/11458/5578/1/Tesis%20Maria%20Aurolina%20Ccarhuaslla%20Ochoa%20ok.pdf>
- Markoulidakis, I., & Markoulidakis, G. (2024). Probabilistic Confusion Matrix: A Novel Method for Machine Learning Algorithm Generalized Performance Analysis. *Technologies*, 12(7), 113. <https://doi.org/10.3390/technologies12070113>
- MIDAGRI. (2024). *MIDAGRI: Incremento del precio del cacao beneficia a productores peruanos*. Ministerio de Desarrollo Agrario y Riego.
- Mohammad Yazdi Pusadan, Syahrullah, Merry, & Ahmad Imam Abdullah. (2022). k-Nearest Neighbor and Feature Extraction on Detection of Pest and Diseases of Cocoa. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(3), 471–480. <https://doi.org/10.29207/resti.v6i3.4064>
- Morales, M., Morocho, J., López, X., & Navas, P. (2024). *Application of Convolutional Neural Networks for the Detection of Diseases in the CCN-51 Cocoa Fruit by Means of a Mobile Application* (pp. 3–10). [https://doi.org/10.1007/978-3-031-47100-1\\_1](https://doi.org/10.1007/978-3-031-47100-1_1)

- Oliveira, R. C. de, & Silva, R. D. de S. e. (2023). Artificial Intelligence in Agriculture: Benefits, Challenges, and Trends. *Applied Sciences*, 13(13), 7405. <https://doi.org/10.3390/app13137405>
- Plasencia, A., Vilchez, C., Ferrer, Y., & Veloz, C. (2022). Efecto del cambio climático sobre la distribución potencial del hongo *Moniliophthora roreri* y el cultivo de cacao (*Theobroma cacao*) en Ecuador continental. *REVISTA TERRA LATINOAMERICANA*, 40. <https://doi.org/10.28940/terra.v40i0.1151>
- Reyes, B. M. D., Fonseca, P. L. C., Heming, N. M., Conceição, L. B. de A., Nascimento, K. T. de S., Gramacho, K. P., Arevalo-Gardini, E., Pirovani, C. P., & Aguiar, E. R. G. R. (2023). Characterization of the microbiota dynamics associated with *Moniliophthora roreri*, causal agent of cocoa frosty pod rot disease, reveals new viral species. *Frontiers in Microbiology*, 13. <https://doi.org/10.3389/fmicb.2022.1053562>
- Rodríguez-Delzo, E. E., Huerta-Guillén, R., & Montañez-Artica, A. G. (2023). Identificación de variables que mejoren la cadena productiva de cacao (*Theobroma cacao* L.) y su importancia para el desarrollo del VRAEM – Perú. *Revista Agrotecnológica Amazónica*, 3(2), e465. <https://doi.org/10.51252/raa.v3i2.465>
- Santa María Pinedo, J. C., Ríos López, C. A., Rodríguez Grández, C., & García Estrella, C. W. (2021). Reconocimiento de patrones de imágenes a través de un sistema de visión artificial en MATLAB. *Revista Científica de Sistemas e Informática*, 1(2), 15–26. <https://doi.org/10.51252/rcsi.v1i2.131>
- Soto Chochocca, R. R., Gonzales Avila, E., Fernandez Rojas, J. H., Angeles Suazo, J. M., Huamán De La Cruz, A. R., & Hadi Mohamed, M. M. (2022). Antifungal effect from *Zingiber officinale*, *Aloe vera* and *Trichoderma* sp. for control of *Moniliophthora roreri* in *Theobroma cacao* in Huánuco, Peru. *Revista Facultad Nacional de Agronomía Medellín*, 75(1). <https://doi.org/10.15446/rfnam.v75n1.95804>
- Suh, N. N., & Molua, E. L. (2022). Cocoa production under climate variability and farm management challenges: Some farmers' perspective. *Journal of Agriculture and Food Research*, 8, 100282. <https://doi.org/10.1016/j.jafr.2022.100282>
- Valenzuela-Cobos, J. D., Guevara-Viejó, F., Vicente-Galindo, P., & Galindo-Villardón, P. (2023). Eco-Friendly Biocontrol of Moniliasis in Ecuadorian Cocoa Using Biplot Techniques. *Sustainability*, 15(5), 4223. <https://doi.org/10.3390/su15054223>

- Villalobos-Culqui, C., Valles-Coral, M. A., García-Rivas-Plata, C., & Tuesta-Hidalgo, O. A. (2025). Modelo de visión artificial basada en redes neuronales convolucionales para identificación de mazorca negra en plantaciones de cacao. *Revista Científica de Sistemas e Informática*, 5(1), e678. <https://doi.org/10.51252/rcsi.v5i1.678>
- Wongnaa, C. A., Jelilu, F., Apike, I. A., Djokoto, J. G., & Awunyo-Vitor, D. (2022). Effect of hybrid cocoa seedlings adoption on profit efficiency in Ghana. *Sustainable Futures*, 4, 100074. <https://doi.org/10.1016/j.sftr.2022.100074>

## ANEXOS

## Anexo 1: Autorización para visita de parcelas del ICT

---

### Solicitud de permiso para visita al Instituto de Cultivos Tropicales (ICT)

---

Instituto de Cultivos Tropicales <ict.org.peru@gmail.com>

5 de febrero de 2025, 9:00 a.m.

Para: Jhosep Sánchez Flores <jhosepsanfio@gmail.com>

CC: Cesar Arevalo <cesar.arevaloh@gmail.com>, ctenazoasanchez0310@gmail.com

Estimado:Josep

Buenos días, agendamos su visita para la fecha que usted menciona, Jueves 06 de febrero a las **7:30 am** , El encargado de su atención será el **ing. Carlos Mauro Tenazoa** , coordinador de campo del ICT.

Saludos..

Ronald Garcia Garcia

**Atencion al Cliente - Instituto de Cultivos Tropicales**

**Celular : 942663941**

## Anexo 2: Constancia de Revisión de Imágenes

### CONSTANCIA DE REUNIÓN DE TRABAJO

**Lugar :** Instituto de Cultivos Tropicales - La Banda de Shilcayo - San Martín - San Martín

**Fecha:** 03/06/2025

**Hora de inicio:** 03:00pm

**Reunidos:**

- Diego Emmanuel Rengifo Lozano, identificado con DNI N.º 72803521, en calidad de investigador responsable del proyecto: "Evaluación de modelos híbridos basado en extracción de característica y regresión para clasificación de moniliasis en imágenes de cacao".

- Carlos Mauro Tenazoa, identificado con DNI N.º 47385308, con título profesional de Ingeniero Agrónomo, en calidad de especialista en fitopatología.

**Objetivo de la reunión:**

Coordinar y validar el proceso de etiquetado de imágenes de frutos de cacao que presenten síntomas de la enfermedad Monilia, así como de frutos sanos, con el fin de construir un conjunto de datos confiable para el entrenamiento de modelos de inteligencia artificial.

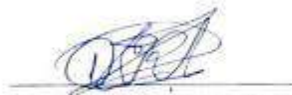
**Actividades a realizar:**

El ingeniero agrónomo brinda su asesoría técnica y valida las etiquetas correspondientes a cada clase que se presente en las imágenes proporcionadas por el investigador.

Revisión de las imágenes recolectadas y clasificación de cada una según la clase presente (Monilia o Sana), de acuerdo con los criterios del especialista.

La participación del ingeniero agrónomo será registrada y reconocida como parte del proceso de validación del conjunto de datos del estudio.

**Firman en señal de conformidad:**



Diego Emmanuel Rengifo Lozano  
Investigador responsable



Carlos Mauro Tenazoa  
Ingeniero Agrónomo

## Anexo 3: Código fuente

Link: <https://colab.research.google.com/drive/1C4j6bf0LjD-DnQXaq0BNUeGcWL6VGePP?usp=sharing>

```
#Instalar librerías
!pip install rembg
!pip install rembg onnxruntime

# Librerías estándar y utilitarias
import os
import io
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm

# Google Colab
from google.colab import drive

# Procesamiento de imágenes
from PIL import Image, ImageOps
from rembg import remove

# TensorFlow y Keras
import tensorflow as tf
from tensorflow.keras import layers, models, Input, regularizers
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array,
load_img, array_to_img
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLRonPlateau

# Modelado y evaluación con Scikit-learn
from sklearn.model_selection import StratifiedShuffleSplit, RandomizedSearchCV,
cross_validate, train_test_split, StratifiedKFold, KFold, cross_val_predict
from sklearn.preprocessing import StandardScaler, label_binarize
import seaborn as sns
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score,
mean_absolute_error, r2_score, roc_auc_score, matthews_corrcoef, confusion_matrix,
ConfusionMatrixDisplay, make_scorer, roc_curve, auc, mean_squared_error

# Modelos de regresión
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from xgboost import XGBRegressor

# Estadística
from IPython.display import display
from scipy.stats import uniform, randint, f_oneway
from statsmodels.stats.multicomp import pairwise_tukeyhsd

#Conectar a Drive
drive.mount('/content/drive')

#Mostrar dataset
def mostrar_muestras(ruta_clase, titulo):
    imagenes = os.listdir(ruta_clase)[:25]
    fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(5,5))
    fig.suptitle(titulo, fontsize=16, y=0.97)
    axs = axs.flatten()
    for i, nombre_img in enumerate(imagenes):
        img_path = os.path.join(ruta_clase, nombre_img)
        img = Image.open(img_path).resize((224, 224)).rotate(270)
        axs[i].imshow(img)
        axs[i].axis('off')
    for j in range(i + 1, 15):
        axs[j].axis('off')
    plt.tight_layout()
    plt.subplots_adjust(wspace=0, hspace=0.05, top=0.90)
    plt.show()
mostrar_muestras('/content/drive/MyDrive/Dataset Cacao/Sanos', 'Sano')
mostrar_muestras('/content/drive/MyDrive/Dataset Cacao/Monilia', 'Monilia')

#Balanceo de datos
```

```
def cantidad_imagenes(ruta_base):
    clases = ['Monilia', 'Sanos']
    cantidades = []
    for clase in clases:
        ruta_clase = os.path.join(ruta_base, clase)
        imagenes = [f for f in os.listdir(ruta_clase) if f.lower().endswith(('.jpg',
        '.jpeg', '.png'))]
        cantidades.append(len(imagenes))

    # Crear gráfico de barras
    plt.figure(figsize=(5,5))
    barras = plt.bar(clases, cantidades, color=['red', 'green'])

    # Añadir etiquetas encima
    for bar in barras:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2.0, yval + 2, str(yval), ha='center',
va='bottom', fontsize=10)
    plt.title('Cantidad de imágenes por clase', fontsize=14)
    plt.ylim(0, max(cantidades) + 100)
    plt.grid(axis='y', linestyle='--', alpha=0.5)
    plt.tight_layout()
    plt.show()
cantidad_imagenes('/content/drive/MyDrive/Dataset Cacao')

#Aumento de datos
#Clase monilia
# Configuración
carpeta_origen = '/content/drive/MyDrive/Dataset Cacao/Monilia'
carpeta_destino = '/content/drive/MyDrive/Dataset Cacao Aumento Datos/Monilia'
os.makedirs(carpeta_destino, exist_ok=True)
objetivo_por_clase = 2000
augmentador = ImageDataGenerator(
    rotation_range=25,
    horizontal_flip=True,
    zoom_range=0.15,
    fill_mode='nearest'
)
imagenes = [f for f in os.listdir(carpeta_origen) if f.lower().endswith(('.jpg',
'.jpeg', '.png'))]
num_originales = len(imagenes)
aumentos_por_imagen = (objetivo_por_clase - num_originales) // num_originales
residuo = (objetivo_por_clase - num_originales) % num_originales
contador = 1
residuo_agregado = 0
for archivo in tqdm(imagenes, desc='Aumentando clase Monilia'):
    ruta_img = os.path.join(carpeta_origen, archivo)
    ext = os.path.splitext(archivo)[1].lower() # .jpg, .png
    img = Image.open(ruta_img)
    img = ImageOps.exif_transpose(img) #  Corrige orientación EXIF
    img_array = img_to_array(img)
    #  Guardar la imagen original
    img_pil = array_to_img(img_array.astype('uint8'))
    ruta_original = os.path.join(carpeta_destino, f'monilia_{contador}{ext}')
    if ext in ['.jpg', '.jpeg']:
        img_pil.save(ruta_original, format='JPEG', quality=100, subsampling=0)
    else:
        img_pil.save(ruta_original, format='PNG')
    contador += 1
    #  Aumentos básicos
    for _ in range(aumentos_por_imagen):
        img_aug = augmentador.random_transform(img_array)
        img_pil = array_to_img(img_aug.astype('uint8'))

        ruta_guardado = os.path.join(carpeta_destino, f'monilia_{contador}{ext}')
        if ext in ['.jpg', '.jpeg']:
            img_pil.save(ruta_guardado, format='JPEG', quality=100, subsampling=0)
        else:
            img_pil.save(ruta_guardado, format='PNG')
        contador += 1
    #  Aumentos extra si hay residuo
    if residuo_agregado < residuo:
        img_aug = augmentador.random_transform(img_array)
        img_pil = array_to_img(img_aug.astype('uint8'))

        ruta_guardado = os.path.join(carpeta_destino, f'monilia_{contador}{ext}')
```

```
        if ext in ['.jpg', '.jpeg']:
            img_pil.save(ruta_guardado, format='JPEG', quality=100, subsampling=0)
        else:
            img_pil.save(ruta_guardado, format='PNG')
        contador += 1
        residuo_agregado += 1
archivos_finales = [f for f in os.listdir(carpeta_destino) if
f.lower().endswith(('.jpg', '.jpeg', '.png'))]
print(f"\n✅ Total de imágenes: {len(archivos_finales)}")

#Sanos
# Configuración
carpeta_origen = '/content/drive/MyDrive/Dataset Cacao/Sanos'
carpeta_destino = '/content/drive/MyDrive/Dataset Cacao Aumento Datos/Sanos'
os.makedirs(carpeta_destino, exist_ok=True)
objetivo_por_clase = 2000
augmentador = ImageDataGenerator(
    rotation_range=25,
    horizontal_flip=True,
    zoom_range=0.15,
    fill_mode='nearest'
)
imagenes = [f for f in os.listdir(carpeta_origen) if f.lower().endswith(('.jpg',
'.jpeg', '.png'))]
num_originales = len(imagenes)
aumentos_por_imagen = (objetivo_por_clase - num_originales) // num_originales
residuo = (objetivo_por_clase - num_originales) % num_originales
contador = 1
residuo_agregado = 0
for archivo in tqdm(imagenes, desc='Aumentando clase Sanos'):
    ruta_img = os.path.join(carpeta_origen, archivo)
    ext = os.path.splitext(archivo)[1].lower() # .jpg, .png
    img = Image.open(ruta_img)
    img = ImageOps.exif_transpose(img) # ✅ Corrige orientación EXIF
    img_array = img_to_array(img)
    # ✅ Guardar la imagen original
    img_pil = array_to_img(img_array.astype('uint8'))
    ruta_original = os.path.join(carpeta_destino, f'sano_{contador}{ext}')
    if ext in ['.jpg', '.jpeg']:
        img_pil.save(ruta_original, format='JPEG', quality=100, subsampling=0)
    else:
        img_pil.save(ruta_original, format='PNG')
    contador += 1
    # 📦 Aumentos básicos
    for _ in range(aumentos_por_imagen):
        img_aug = augmentador.random_transform(img_array)
        img_pil = array_to_img(img_aug.astype('uint8'))
        ruta_guardado = os.path.join(carpeta_destino, f'sano_{contador}{ext}')
        if ext in ['.jpg', '.jpeg']:
            img_pil.save(ruta_guardado, format='JPEG', quality=100, subsampling=0)
        else:
            img_pil.save(ruta_guardado, format='PNG')
        contador += 1
    # 📦 Aumentos extra si hay residuo
    if residuo_agregado < residuo:
        img_aug = augmentador.random_transform(img_array)
        img_pil = array_to_img(img_aug.astype('uint8'))
        ruta_guardado = os.path.join(carpeta_destino, f'sano_{contador}{ext}')
        if ext in ['.jpg', '.jpeg']:
            img_pil.save(ruta_guardado, format='JPEG', quality=100, subsampling=0)
        else:
            img_pil.save(ruta_guardado, format='PNG')
        contador += 1
        residuo_agregado += 1
archivos_finales = [f for f in os.listdir(carpeta_destino) if
f.lower().endswith(('.jpg', '.jpeg', '.png'))]
print(f"\n✅ Total de imágenes: {len(archivos_finales)}")
cantidad_imagenes('/content/drive/MyDrive/Dataset Cacao Aumento Datos')
#Segmentacion de imagenes clase sano
carpeta_entrada = '/content/drive/MyDrive/Dataset Cacao Aumento Datos/Sanos'
carpeta_salida = '/content/drive/MyDrive/Dataset Cacao Segmentado/Sanos'

imagenes = [f for f in os.listdir(carpeta_entrada) if f.lower().endswith(('.png',
'.jpg', '.jpeg'))]
```

```
for i, archivo in enumerate(imagenes, start=1):
    ruta_imagen = os.path.join(carpeta_entrada, archivo)

    with open(ruta_imagen, 'rb') as f:
        input_img = f.read()
        output_img = remove(input_img)

        img_sin_fondo = Image.open(io.BytesIO(output_img)).convert("RGBA")
        img_sin_fondo = img_sin_fondo.resize((224, 224), Image.LANCZOS)

        nombre_salida = f"sano_{i}.png"
        ruta_salida = os.path.join(carpeta_salida, nombre_salida)

        img_sin_fondo.save(ruta_salida, format="PNG")
        print(f"✅ ({i}/{len(imagenes)}) Guardado: {nombre_salida}")

#Segmentacion de imagenes clase monilia
carpeta_entrada = '/content/drive/MyDrive/Dataset Cacao Aumento Datos/Monilia'
carpeta_salida = '/content/drive/MyDrive/Dataset Cacao Segmentado/Monilia'

imagenes = [f for f in os.listdir(carpeta_entrada) if f.lower().endswith(('.png',
'.jpg', '.jpeg'))]

for i, archivo in enumerate(imagenes, start=1):
    ruta_imagen = os.path.join(carpeta_entrada, archivo)

    with open(ruta_imagen, 'rb') as f:
        input_img = f.read()
        output_img = remove(input_img)

        img_sin_fondo = Image.open(io.BytesIO(output_img)).convert("RGBA")
        img_sin_fondo = img_sin_fondo.resize((224, 224), Image.LANCZOS)
        nombre_salida = f"monilia_{i}.png"
        ruta_salida = os.path.join(carpeta_salida, nombre_salida)

        img_sin_fondo.save(ruta_salida, format="PNG")
        print(f"({i}/{len(imagenes)}) Guardado: {nombre_salida}")

#Mostrar imagenes segmentadas
def mostrar_muestras_seg(ruta_clase, titulo):
    imagenes = os.listdir(ruta_clase)[:25]
    fig, axs = plt.subplots(nrows=5, ncols=5, figsize=(5,5))
    fig.suptitle(titulo, fontsize=16, y=0.97)

    axs = axs.flatten()

    for i, nombre_img in enumerate(imagenes):
        img_path = os.path.join(ruta_clase, nombre_img)
        img = Image.open(img_path).resize((224, 224)).rotate(270)
        axs[i].imshow(img)
        axs[i].axis('off')

    for j in range(i + 1, 15):
        axs[j].axis('off')

    plt.tight_layout()
    plt.subplots_adjust(wspace=0, hspace=0.05, top=0.90)
    plt.show()

mostrar_muestras_seg('/content/drive/MyDrive/Dataset Cacao Segmentado/Sanos', 'Sano')
mostrar_muestras_seg('/content/drive/MyDrive/Dataset Cacao Segmentado/Monilia',
'Monilia')

#Carga de imagenes y almacen de datos en array
ruta_base = '/content/drive/MyDrive/Dataset Cacao Segmentado'
clases = {'Sanos': 0, 'Monilia': 1}
tamaño_img = (224, 224)
X, y = [], []
for clase, etiqueta in clases.items():
    carpeta = os.path.join(ruta_base, clase)
    imagenes = [f for f in os.listdir(carpeta) if f.lower().endswith(('.png', '.jpg',
'.jpeg'))]
    for nombre in tqdm(imagenes, desc=f'Procesando {clase}'):
        ruta_img = os.path.join(carpeta, nombre)
        img = Image.open(ruta_img).convert("RGB").resize(tamaño_img)
        img_array = np.array(img)
        img_array = preprocess_input(img_array)
```

```
X.append(img_array)
y.append(etiqueta)
X = np.array(X)
y = np.array(y).astype(np.float32)

# Separar dataset para fine-tuning
X_finetune, X_regresores, y_finetune, y_regresores = train_test_split(
    X, y, test_size=0.75, stratify=y, random_state=42)

# Preparar datasets para fine-tuning
dataset_ft = tf.data.Dataset.from_tensor_slices((X_finetune,
y_finetune)).shuffle(len(X_finetune), seed=42)
train_size = int(0.8 * len(X_finetune))
train_ds = dataset_ft.take(train_size).batch(32)
val_ds = dataset_ft.skip(train_size).batch(32)

# Hiperparámetros a buscar
capas_a_probar = [20, 30, 40]
learning_rates = [1e-4, 1e-5]
dropouts = [0.3, 0.5]
mejor_val_loss = float('inf')
mejor_modelo = None
mejor_config = {}
mejor_history = None
for n_capas in capas_a_probar:
    for lr in learning_rates:
        for dr in dropouts:
            print(f"Probando: capas={n_capas}, lr={lr}, dropout={dr}")
            base_model = MobileNetV2(input_shape=(224, 224, 3), include_top=False,
weights='imagenet', pooling='avg')
            for layer in base_model.layers[:-n_capas]:
                layer.trainable = False
            inputs = Input(shape=(224, 224, 3))
            x = base_model(inputs)
            x = layers.Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.001))(x)
            x = layers.Dropout(dr)(x)
            outputs = layers.Dense(1, activation='linear')(x)

            model = models.Model(inputs, outputs)
            model.compile(optimizer=Adam(learning_rate=lr), loss='mse', metrics=['mae'])

            early_stop = EarlyStopping(monitor='val_loss', patience=3,
restore_best_weights=True)
            reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2)

            history = model.fit(train_ds, validation_data=val_ds, epochs=20,
callbacks=[early_stop, reduce_lr], verbose=0)

            val_loss = history.history['val_loss'][-1]
            if val_loss < mejor_val_loss:
                mejor_val_loss = val_loss
                mejor_modelo = model
                mejor_config = {'capas': n_capas, 'lr': lr, 'dropout': dr}
                mejor_history = history
print("Mejor configuración:", mejor_config)

#  Graficar val_loss del mejor configuración
plt.figure(figsize=(8, 5))
plt.plot(mejor_history.history['val_loss'], marker='o')
plt.title('Pérdida de validación (val_loss) - Mejor Configuración')
plt.xlabel('Época')
plt.ylabel('val_loss')
plt.grid(True)
plt.tight_layout()
plt.show()

# Curva de aprendizaje
plt.figure(figsize=(8, 5))
plt.plot(mejor_history.history['loss'], label='Entrenamiento')
plt.plot(mejor_history.history['val_loss'], label='Validación')
plt.xlabel("Época")
plt.ylabel("Loss (MSE)")
plt.title("Curva de Aprendizaje - Fine-tuning MobileNetV2")
plt.legend()
plt.grid(True)
```

```
plt.tight_layout()
plt.show()

#Extracción de características
feature_model = tf.keras.Model(inputs=mejor_modelo.input, outputs=mejor_modelo.layers[-2].output)
sss = StratifiedShuffleSplit(n_splits=1, test_size=476, random_state=42)
for train_idx, eval_idx in sss.split(X_regresores, y_regresores):
    X_train_regresores = X_regresores[train_idx]
    y_train_regresores = y_regresores[train_idx]
    X_eval_final = X_regresores[eval_idx]
    y_eval_final = y_regresores[eval_idx]

# Extracción de características con modelo ya fine-tuneado
X_features_train = feature_model.predict(X_train_regresores, batch_size=32, verbose=1)
X_features_eval = feature_model.predict(X_eval_final, batch_size=32, verbose=1)

# Guardar características y etiquetas
np.save("/content/drive/MyDrive/features_train_moniliasis.npy", X_features_train)
np.save("/content/drive/MyDrive/labels_train_moniliasis.npy", y_train_regresores)

np.save("/content/drive/MyDrive/features_eval_moniliasis.npy", X_features_eval)
np.save("/content/drive/MyDrive/labels_eval_moniliasis.npy", y_eval_final)

# Combinar features y etiquetas en un DataFrame
df_features = pd.DataFrame(X_features_train)
df_features['label'] = y_train_regresores # Añade la columna de etiquetas

df_featuresEva = pd.DataFrame(X_features_eval)
df_featuresEva['label'] = y_eval_final # Añade la columna de etiquetas

# Guardar como CSV en Drive
csv_path = "/content/drive/MyDrive/features_train_moniliasis.csv"
df_features.to_csv(csv_path, index=False)

csv_pathEva = "/content/drive/MyDrive/features_eval_moniliasis.csv"
df_featuresEva.to_csv(csv_pathEva, index=False)

X_features = np.load("/content/drive/MyDrive/features_train_moniliasis.npy")
y_final = np.load("/content/drive/MyDrive/labels_train_moniliasis.npy")
# Escalar características para SVR y KNN
from sklearn.pipeline import make_pipeline
#Encontrar mejores hiperparametros con randomSearch
# ---1. SVR---
svr_pipe = make_pipeline(StandardScaler(), SVR())
param_svr = {
    'svr_C': np.logspace(-1, 2, 10),
    'svr_epsilon': np.linspace(0.01, 0.2, 10),
    'svr_gamma': ['scale', 'auto']
}
random_search_svr = RandomizedSearchCV(
    estimator=svr_pipe,
    param_distributions=param_svr,
    n_iter=20,
    cv=5,
    scoring='neg_mean_absolute_error',
    verbose=2,
    random_state=42,
    n_jobs=-1
)
random_search_svr.fit(X_features, y_final)
print("Mejores hiperparámetros SVR:", random_search_svr.best_params_)

# === 2. KNeighborsRegressor ===
knn_pipe = make_pipeline(StandardScaler(), KNeighborsRegressor())
param_knn = {
    'kneighborsregressor_n_neighbors': list(range(3, 20)),
    'kneighborsregressor_weights': ['uniform', 'distance'],
    'kneighborsregressor_p': [1, 2]
}
random_search_knn = RandomizedSearchCV(
    estimator=knn_pipe,
    param_distributions=param_knn,
    n_iter=20,
    cv=5,
    scoring='neg_mean_absolute_error',
    verbose=2,
```

```
    random_state=42,
    n_jobs=-1
)
random_search_knn.fit(X_features, y_final)
print("Mejores hiperparámetros KNN:", random_search_knn.best_params_)

# === 3. XGBRegressor ===
xgb = XGBRegressor(objective='reg:squarederror', random_state=42)
param_xgb = {
    'n_estimators': [50, 100, 150, 200],
    'max_depth': [2, 3, 4, 5],
    'learning_rate': np.linspace(0.01, 0.3, 10),
    'subsample': [0.7, 0.8, 1.0]
}
random_search_xgb = RandomizedSearchCV(
    estimator=xgb,
    param_distributions=param_xgb,
    n_iter=20,
    cv=5,
    scoring='neg_mean_absolute_error',
    verbose=2,
    random_state=42,
    n_jobs=-1
)
random_search_xgb.fit(X_features, y_final)
print("Mejores hiperparámetros XGBoost:", random_search_xgb.best_params_)
kf = KFold(n_splits=5, shuffle=True, random_state=42)
resumen = []
modelo_svr = random_search_svr.best_estimator_

# --- Predicción cruzada (regresor continuo) ---
y_pred_svr = cross_val_predict(modelo_svr, X_features, y_final, cv=kf)
residuos = y_final - y_pred_svr

# === GRÁFICOS antes de binarizar ===

# 1. Dispersión: y_real vs y_pred
plt.figure(figsize=(6, 4))
plt.scatter(y_final, y_pred_svr, alpha=0.6)
plt.plot([0, 1], [0, 1], 'r--')
plt.title("Dispersión: Real vs Predicho (MOBILENETV2 - SVR)")
plt.xlabel("Valor Real")
plt.ylabel("Valor Predicho")
plt.grid(True)
plt.tight_layout()
plt.show()

# 2. Histograma de residuos
plt.figure(figsize=(6, 4))
sns.histplot(residuos, kde=True, bins=20)
plt.title("Distribución de Residuos MOBILENETV2 - SVR")
plt.xlabel("Error (Real - Predicho)")
plt.tight_layout()
plt.show()

# 3. Predicciones ordenadas
plt.figure(figsize=(6, 4))
plt.plot(sorted(y_final), label='Real', marker='o')
plt.plot(sorted(y_pred_svr), label='Predicho', marker='x')
plt.title("Predicciones Ordenadas MOBILENETV2 - SVR")
plt.xlabel("Índice ordenado")
plt.ylabel("Valor")
plt.legend()
plt.tight_layout()
plt.show()

# === Binarización ===
y_pred_bin = (y_pred_svr >= 0.5).astype(int)
class_names = ['Sano', 'Monilia']

# === Métricas ===
acc_svr = accuracy_score(y_final, y_pred_bin)
prec_svr = precision_score(y_final, y_pred_bin)
rec_svr = recall_score(y_final, y_pred_bin)
f1_svr = f1_score(y_final, y_pred_bin)
mcc_svr = matthews_corrcoef(y_final, y_pred_bin)
mse_svr = mean_squared_error(y_final, y_pred_svr)
```

```
auc_svr = roc_auc_score(y_final, y_pred_svr)

# === ROC por clase ===
y_bin = label_binarize(y_final, classes=[0, 1])
if y_bin.shape[1] == 1:
    y_bin = np.hstack([1 - y_bin, y_bin])
y_pred_multi = np.vstack((1 - y_pred_svr, y_pred_svr)).T

plt.figure(figsize=(7, 5))
for i, name in enumerate(class_names):
    fpr_i, tpr_i, _ = roc_curve(y_bin[:, i], y_pred_multi[:, i])
    auc_i = auc(fpr_i, tpr_i)
    plt.plot(fpr_i, tpr_i, label=f"ROC Clase {name} (AUC = {auc_i:.4f})")

plt.plot([0, 1], [0, 1], 'k--')
plt.title("Curva ROC por clase MOBILENETV2 - SVR")
plt.xlabel("Falso Positivo")
plt.ylabel("Verdadero Positivo")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# === Matriz de Confusión ===
plt.figure(figsize=(5, 4))
sns.heatmap(confusion_matrix(y_final, y_pred_bin), annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names)
plt.title("Matriz de Confusión MOBILENETV2 - SVR")
plt.xlabel("Predicho")
plt.ylabel("Real")
plt.tight_layout()
plt.show()

# === Resumen ===
resumen_svr=[]
resumen_svr.append({
    'Modelo': 'Mobilenetv2 - SVR',
    'Accuracy (%)': acc_svr * 100,
    'Precision (%)': prec_svr * 100,
    'Recall (%)': rec_svr * 100,
    'F1 Score (%)': f1_svr * 100,
    'MCC (%)': mcc_svr * 100,
    'MSE': mse_svr,
    'AUC': auc_svr
})
df_resumen = pd.DataFrame(resumen_svr)
display(df_resumen)
mejor_knn = random_search_knn.best_estimator_

# Validación cruzada
kf = KFold(n_splits=5, shuffle=True, random_state=42)
y_pred_knn = cross_val_predict(mejor_knn, X_features, y_final, cv=kf)
y_pred_knn_bin = (y_pred_knn >= 0.5).astype(int)

# Métricas
mse_knn = mean_squared_error(y_final, y_pred_knn)
acc_knn = accuracy_score(y_final, y_pred_knn_bin)
prec_knn = precision_score(y_final, y_pred_knn_bin)
rec_knn = recall_score(y_final, y_pred_knn_bin)
f1_knn = f1_score(y_final, y_pred_knn_bin)
mcc_knn = matthews_corrcoef(y_final, y_pred_knn_bin)
auc_knn = roc_auc_score(y_final, y_pred_knn)

# === Gráfico 1: Dispersión ===
plt.figure(figsize=(6, 5))
plt.scatter(y_final, y_pred_knn, alpha=0.4)
plt.plot([0, 1], [0, 1], 'r--')
plt.title("Dispersión: Real vs Predicho (MOBILENETV2 - KNN)")
plt.xlabel("Valor Real")
plt.ylabel("Valor Predicho")
plt.grid(True)
plt.tight_layout()
plt.show()

# === Gráfico 2: Histograma de predicciones ===
plt.figure(figsize=(6, 4))
sns.histplot(y_pred_knn, kde=True, bins=20, color='orange', edgecolor='black')
```

```
plt.title("Distribución de probabilidades predichas (MOBILENETV2 - KNN)")
plt.xlabel("Valor predicho")
plt.ylabel("Frecuencia")
plt.grid(True)
plt.tight_layout()
plt.show()

# === Gráfico 3: Distribución de residuos ===
errores = y_final - y_pred_knn
plt.figure(figsize=(6, 4))
sns.histplot(errores, kde=True, bins=20, color='skyblue')
plt.title("Distribución de Residuos MOBILENETV2 - KNN")
plt.xlabel("Error (Real - Predicho)")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.show()

# Binarizar etiquetas reales (one-hot)
y_bin = label_binarize(y_final, classes=[0, 1])
if y_bin.shape[1] == 1:
    y_bin = np.hstack([1 - y_bin, y_bin]) # Asegura forma (n, 2)

# Crear predicciones como probabilidad por clase
y_pred_multi = np.vstack((1 - y_pred_knn, y_pred_knn)).T

# Curva ROC por clase
plt.figure(figsize=(7, 5))
for i, name in enumerate(class_names):
    fpr_i, tpr_i, _ = roc_curve(y_bin[:, i], y_pred_multi[:, i])
    auc_i = roc_auc_score(y_bin[:, i], y_pred_multi[:, i])
    plt.plot(fpr_i, tpr_i, label=f"ROC clase {name} (AUC = {auc_i:.4f})")

plt.plot([0, 1], [0, 1], 'k--')
plt.title("Curva ROC por clase MOBILENETV2 - KNN")
plt.xlabel("Falso Positivo")
plt.ylabel("Verdadero Positivo")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# === Matriz de Confusión ===
plt.figure(figsize=(5, 4))
sns.heatmap(confusion_matrix(y_final, y_pred_knn_bin), annot=True, fmt='d',
cmap='Blues',
xticklabels=['Sano', 'Monilia'], yticklabels=['Sano', 'Monilia'])
plt.title("Matriz de Confusión MOBILENETV2 - KNN")
plt.xlabel("Etiqueta predicha")
plt.ylabel("Etiqueta real")
plt.tight_layout()
plt.show()

# === Resumen ===
resumen_knn = pd.DataFrame([
    'Modelo': 'Mobilenetv2 - KNN',
    'Accuracy (%)': acc_knn * 100,
    'Precision (%)': prec_knn * 100,
    'Recall (%)': rec_knn * 100,
    'F1 Score (%)': fl_knn * 100,
    'MCC (%)': mcc_knn * 100,
    'MSE': mse_knn,
    'AUC': auc_knn
])
df_resumen = pd.DataFrame(resumen_knn)
display(df_resumen)
# Usamos el mejor estimador encontrado
modelo_xgb = random_search_xgb.best_estimator_

# Validación cruzada con k=5
y_pred_xgb = cross_val_predict(modelo_xgb, X_features, y_final, cv=5, n_jobs=-1)
mse_xgb = mean_squared_error(y_final, y_pred_xgb)

# === Gráficos de regresión antes de binarizar ===

# 1. Gráfico de dispersión real vs predicho
```

```
plt.figure(figsize=(6, 5))
plt.scatter(y_final, y_pred_xgb, alpha=0.4)
plt.plot([0, 1], [0, 1], 'r--')
plt.title("Dispersión real vs predicho MOBILENETV2 - XGBoost")
plt.xlabel("Valor real")
plt.ylabel("Valor predicho")
plt.grid(True)
plt.tight_layout()
plt.show()

# 2. Histograma de probabilidades predichas
plt.figure(figsize=(6, 4))
sns.histplot(y_pred_xgb, kde=True, bins=20, color='orange')
plt.title("Distribución de probabilidades predichas MOBILENETV2 - XGBoost")
plt.xlabel("Probabilidad de clase Monilia")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.show()

# === Binarización ===
y_pred_bin = (y_pred_xgb >= 0.5).astype(int)
class_names = ['Sano', 'Monilia']

# Métricas después de binarizar
fpr_xgb, tpr_xgb, _ = roc_curve(y_final, y_pred_xgb)
auc_xgb = roc_auc_score(y_final, y_pred_xgb)
acc_xgb = accuracy_score(y_final, y_pred_bin)
prec_xgb = precision_score(y_final, y_pred_bin)
rec_xgb = recall_score(y_final, y_pred_bin)
f1_xgb = f1_score(y_final, y_pred_bin)
mcc_xgb = matthews_corrcoef(y_final, y_pred_bin)

# === Curva ROC por clase ===
y_bin = label_binarize(y_final, classes=[0, 1])
if y_bin.shape[1] == 1:
    y_bin = np.hstack([1 - y_bin, y_bin])
y_pred_multi = np.vstack((1 - y_pred_xgb, y_pred_xgb)).T

plt.figure(figsize=(7, 5))
for i, name in enumerate(class_names):
    fpr_i, tpr_i, _ = roc_curve(y_bin[:, i], y_pred_multi[:, i])
    auc_i = roc_auc_score(y_bin[:, i], y_pred_multi[:, i])
    plt.plot(fpr_i, tpr_i, label=f"ROC clase {name} (AUC = {auc_i:.4f})")

plt.plot([0, 1], [0, 1], 'k--')
plt.title("Curva ROC por clase MOBILENETV2 - XGBoost")
plt.xlabel("Falso Positivo")
plt.ylabel("Verdadero Positivo")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# === Matriz de Confusión ===
plt.figure(figsize=(5, 4))
sns.heatmap(confusion_matrix(y_final, y_pred_bin), annot=True, fmt='d', cmap='Blues',
                xticklabels=class_names, yticklabels=class_names)
plt.title("Matriz de Confusión MOBILENETV2 - XGBoost")
plt.xlabel("Etiqueta predicha")
plt.ylabel("Etiqueta real")
plt.tight_layout()
plt.show()

# === Tabla resumen ===
resumen_xgb = pd.DataFrame([
    {'Modelo': 'Mobilenetv2 - XGBoost',
     'Accuracy (%)': acc_xgb * 100,
     'Precision (%)': prec_xgb * 100,
     'Recall (%)': rec_xgb * 100,
     'F1 Score (%)': f1_xgb * 100,
     'MCC (%)': mcc_xgb * 100,
     'MSE': mse_xgb,
     'AUC': auc_xgb
    }])
df_resumen = pd.DataFrame(resumen_xgb)
display(df_resumen)
# Convertir cada resumen
```

```
df_svr = pd.DataFrame(resumen_svr)
df_knn = pd.DataFrame(resumen_knn)
df_xgb = pd.DataFrame(resumen_xgb)

# Combinar los tres DataFrames en uno solo
df_metricas = pd.concat([df_svr, df_knn, df_xgb], ignore_index=True)

# Redondear los valores
df_metricas = df_metricas.round(4)

# Mostrar la tabla
display(df_metricas)
#Evaluacion de modelos
# Cargar datos simulados (se reemplazaría con los .npz reales)
X = np.load("/content/drive/MyDrive/features_eval_moniliasis.npy")
y = np.load("/content/drive/MyDrive/labels_eval_moniliasis.npy")

# Dividir 238 de cada clase en 17 bloques
def dividir_en_bloques(X, y, bloques=17):
    X_sano = X[y == 0]
    X_monilia = X[y == 1]
    y_sano = y[y == 0]
    y_monilia = y[y == 1]
    X_bloques, y_bloques = [], []
    for i in range(bloques):
        X_b = np.concatenate([X_sano[i*14:(i+1)*14], X_monilia[i*14:(i+1)*14]])
        y_b = np.concatenate([y_sano[i*14:(i+1)*14], y_monilia[i*14:(i+1)*14]])
        X_bloques.append(X_b)
        y_bloques.append(y_b)
    return X_bloques, y_bloques

X_bloques, y_bloques = dividir_en_bloques(X, y)

# Gráfico de distribución por bloque
conteos_sano = [np.sum(yb == 0) for yb in y_bloques]
conteos_monilia = [np.sum(yb == 1) for yb in y_bloques]

x = np.arange(1, 18)
width = 0.35

fig, ax = plt.subplots(figsize=(12, 5))
ax.bar(x - width/2, conteos_sano, width, label='Sano', color='red')
ax.bar(x + width/2, conteos_monilia, width, label='Monilia', color='green')

ax.set_xlabel('Bloque')
ax.set_ylabel('Cantidad de imágenes')
ax.set_title('Distribución de imágenes por clase')
ax.set_xticks(x)
ax.legend()
plt.tight_layout()
plt.show()

# === 3. Función para evaluación por ciclo ===
def evaluar_por_bloque(modelo, X_bloques, y_bloques):
    resultados = []
    for i in range(16):
        X_eval = np.concatenate([X_bloques[j] for j in range(17) if j != i])
        y_eval = np.concatenate([y_bloques[j] for j in range(17) if j != i])

        kf = KFold(n_splits=5, shuffle=True, random_state=42)
        y_pred = cross_val_predict(modelo, X_eval, y_eval, cv=kf, n_jobs=-1)
        y_pred_bin = (y_pred >= 0.5).astype(int)

        resultados.append({
            'Ciclo': i+1,
            'Accuracy': accuracy_score(y_eval, y_pred_bin),
            'Precision': precision_score(y_eval, y_pred_bin),
            'Recall': recall_score(y_eval, y_pred_bin),
            'F1 Score': f1_score(y_eval, y_pred_bin),
            'MCC': matthews_corrcoef(y_eval, y_pred_bin),
            'MSE': mean_squared_error(y_eval, y_pred),
            'AUC': roc_auc_score(y_eval, y_pred)
        })
    return pd.DataFrame(resultados).round(4)

# === 4. Modelos a evaluar ===
modelo_svr = random_search_svr.best_estimator_
modelo_knn = random_search_knn.best_estimator_
```

```
modelo_xgb = random_search_xgb.best_estimator_  
  
# === 5. Evaluaciones ===  
resultados_knn = evaluar_por_bloque(modelo_knn, X_bloques, y_bloques)  
# === 6. Mostrar resultados ===  
print("Resultados KNN")  
display(resultados_knn)  
  
resultados_knn.to_excel("/content/drive/MyDrive/resultados_knn.xlsx", index=False)  
print("✅ Resultados de KNN guardados en Drive: resultados_knn.xlsx")  
resultados_svr = evaluar_por_bloque(modelo_svr, X_bloques, y_bloques)  
print("Resultados SVR")  
display(resultados_svr)  
  
resultados_svr.to_excel("/content/drive/MyDrive/resultados_svr.xlsx", index=False)  
print("✅ Resultados de SVR guardados en Drive: resultados_svr.xlsx")  
resultados_xgb = evaluar_por_bloque(modelo_xgb, X_bloques, y_bloques)  
  
print("Resultados XGBoost")  
display(resultados_xgb)  
  
resultados_xgb.to_excel("/content/drive/MyDrive/resultados_xgb.xlsx", index=False)  
print("✅ Resultados de XGBoost guardados en Drive: resultados_xgb.xlsx")
```