



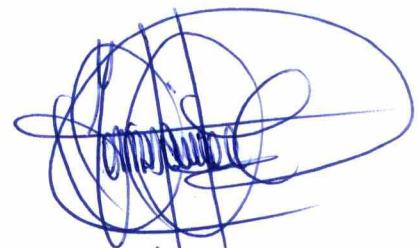
UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO
EN REDES NEURONALES CONVOLUCIONALES PARA LA
DETECCIÓN DEL ESTADO DE MADURACIÓN DEL CACAO
MEDIANTE UNA APLICACIÓN MÓVIL**


Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Adriana Paulina Betancourt Rodríguez
Miguel Ángel Narea Torres
TUTOR: Ing. Jonathan Salvador Paillacho Corredores, M. Sc.

Guayaquil - Ecuador
2025



21/08/2025
11:50



CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Adriana Paulina Betancourt Rodríguez** con documento de identificación N° **0930826102** y **Miguel Ángel Narea Torres** con documento de identificación N° **0959126269**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 21 de agosto del año 2025

Atentamente,



Adriana Paulina Betancourt Rodríguez
0930826102



Miguel Ángel Narea Torres
0959126269

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Adriana Paulina Betancourt Rodríguez** con documento de identificación N° **0930826102** y **Miguel Ángel Narea Torres** con documento de identificación N° **0959126269**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES NEURONALES CONVOLUCIONALES PARA LA DETECCIÓN DEL ESTADO DE MADURACIÓN DEL CACAO MEDIANTE UNA APLICACIÓN MÓVIL**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 21 de agosto del año 2025

Atentamente,



Adriana Paulina Betancourt Rodríguez
0930826102



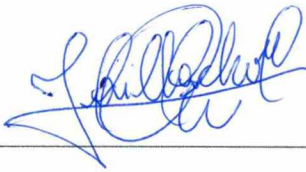
Miguel Ángel Narea Torres
0959126269

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Jonathan Salvador Paillacho Corredores**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES NEURONALES CONVOLUCIONALES PARA LA DETECCIÓN DEL ESTADO DE MADURACIÓN DEL CACAO MEDIANTE UNA APLICACIÓN MÓVIL** , realizado por **Adriana Paulina Betancourt Rodríguez** con documento de identificación N° **0930826102** y por **Miguel Ángel Narea Torres** con documento de identificación N° **0959126269**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 21 de agosto del año 2025

Atentamente,



Ing. Jonathan Salvador Paillacho Corredores. M.Sc.

1718907874

DEDICATORIA

Este trabajo de titulación está dedicado a mi madre Doris Carmen Torres Murillo quien sin su perseverancia y apoyo incondicional no podría ser posible este paso en mi vida que es llegar a ser un profesional, por su responsabilidad y entrega de hacer de mí una persona de bien y valiente para enfrentar cada nuevo reto en mi vida; a mi padre Ángel Manuel Narea Marriott, por su apoyo y sabiduría sin igual para asegurar mi bienestar en el transcurso de todos estos años de estudios universitarios y en mi vida.

También agradezco a Daniela quien fue un apoyo en todo momento y tuve una ayuda importante en toda mi etapa universitaria, finalmente a toda mi familia quienes siempre me han dado sus palabras de aliento, su apoyo moral y quienes nunca dejaron de darme ánimos para terminar esta etapa de mi vida.

Miguel Ángel Narea Torres

Dedico este proyecto de tesis a mi familia, porque gracias a todos ellos he sido capaz de superarme y llegar hasta este punto alto, que yo ví imposible en un principio, pero su devota confianza a través de estos años de estudio, me hicieron ver que sí era capaz. También me gustaría dedicárselo, a todos los amigos que logré hacer en el camino, gracias por permitirme ser parte su vida, y brindarme su apoyo cuando más lo necesité.

Adriana Paulina Betancourt Rodríguez

AGRADECIMIENTO

En primer lugar, deseo expresar mi más sincero agradecimiento a mis padres, quienes han sido el fundamento de mi vida y mi formación. Su esfuerzo, dedicación y sacrificio constante han sido el motor que me impulsó a seguir adelante en cada etapa de este camino académico. Sus valores, su ejemplo de perseverancia y su amor incondicional me han guiado no solo en mis estudios, sino también en mi desarrollo personal.

De igual manera, extendiendo mi gratitud a toda mi familia, quienes con su apoyo, comprensión y palabras de aliento me brindaron la fortaleza necesaria para continuar en los momentos más difíciles. La confianza que depositaron en mí ha sido una motivación invaluable para alcanzar este objetivo.

Este logro no representa únicamente un esfuerzo personal, sino también el resultado del respaldo, la paciencia y el acompañamiento de quienes siempre creyeron en mí.

Miguel Ángel Narea Torres

Agradezco primeramente a Dios, por darme la vida y las fuerzas que necesité cada día para sobrellevar las diferentes dificultades con las que he tenido que enfrentar. Agradezco a mi madre, María Rodríguez Casierra, quien ha sido mi sustento durante toda mi vida, siempre pensando en mi bienestar y en que salga adelante, es la responsable de que lograra llegar hasta aquí, brindándome su apoyo incondicional, en los buenos y malos momentos de la vida, siendo un pilar indispensable en mi trayectoria como estudiante universitaria. Agradezco a mi padre Nelson Betancourt Plúas, quién aunque en su enfermedad, me ha apoyado influenciando sus aprendizajes, demostrándome que cualquier cosa es posible con perseverancia y dedicación. Y finalmente agradezco a mi hermana Isaangie Betancourt Rodríguez, mi confidente de toda la vida, quién me acompañó en los momentos más difíciles, y me dió consejos que me ayudaron a cursar todos estos años universitarios, gracias por ser de gran influencia en mi vida.

Adriana Paulina Betancourt Rodríguez

RESUMEN

Este proyecto describe el diseño e implementación de un sistema inteligente basado en visión artificial para determinar el grado de maduración de las mazorcas de cacao a través de un aplicativo móvil. Dada la importancia del cacao fino de aroma en Ecuador y la necesidad de optimizar los procesos de cosecha, se plantea una solución tecnológica que automatiza la identificación del estado madurativo de las mazorcas directamente en el entorno de cultivo. Para ello, se emplearon imágenes reales capturadas en plantaciones, las cuales fueron clasificadas y etiquetadas con el fin de entrenar un modelo de procesamiento de imágenes.

El entrenamiento del modelo se llevó a cabo bajo un enfoque supervisado, diferenciando entre mazorcas inmaduras, en proceso de maduración y sobre maduras, alcanzando altos niveles de precisión tras varias iteraciones (96 % en promedio). Posteriormente, el modelo fue exportado y optimizado para su integración en un aplicativo web interactivo, que permite a los usuarios cargar, procesarlas localmente y obtener la clasificación visual del estado de maduración de cada mazorca.

Como resultado, se obtuvo un sistema funcional, accesible y adaptable, dirigido a pequeños y medianos productores, que facilita la toma de decisiones sobre el momento óptimo de cosecha, disminuye las pérdidas por recolección prematura o tardía; y promueve la agricultura de precisión en zonas rurales. La metodología propuesta, además de ser aplicable al cacao, puede adaptarse a otros cultivos que requieran análisis visual del fruto, contribuyendo a la digitalización del sector agrícola y fomentando prácticas sostenibles y eficientes en las cadenas de producción.

Palabras clave: visión artificial, etapas de maduración del cacao, aplicativo móvil, procesamiento de imágenes, agricultura de precisión.

ABSTRACT

This project describes the design and implementation of an intelligent system based on computer vision to determine the ripeness stage of cocoa pods through a mobile application. Given the importance of fine aroma cocoa in Ecuador and the need to optimize harvesting processes, a technological solution is proposed to automate the identification of pod maturity stages directly in the cultivation environment. For this purpose, real images captured in plantations were classified and labeled to train an image processing model.

The model training was conducted using a supervised approach, distinguishing between immature, ripening, and mature pods, achieving high levels of accuracy after several iterations (96 % on average). Subsequently, the model was exported and optimized for integration into an interactive web application, allowing users to upload, process them locally, and obtain the visual classification of each pod's maturity stage.

As a result, a functional, accessible, and adaptable system was developed, aimed at small and medium-sized producers, to facilitate decision-making regarding the optimal harvesting time, reduce losses due to premature or late collection; and promote precision agriculture in rural areas. The proposed methodology, in addition to its applicability to cocoa, can be adapted to other crops requiring visual fruit analysis, contributing to the digitalization of the agricultural sector and fostering sustainable and efficient practices within production chains.

Keywords: computer vision, cocoa ripeness, mobile application, image processing, precision agriculture.

ÍNDICE

I.	Introducción	1
II.	Problema	2
III.	Justificación	3
IV.	Objetivos	4
IV-A.	Objetivo general	4
IV-B.	Objetivos específicos	4
V.	Marco teórico	5
V-A.	El cultivo del cacao en Ecuador	5
V-A1.	Características y variedades	5
V-A2.	Estado de madurez del cacao	6
V-A3.	Producción y rendimiento	8
V-A4.	Producción en la Provincia de los Ríos	8
V-A5.	Producción y características de los agricultores	8
V-A6.	Evolución de la producción (2015-2018)	9
V-A7.	Comercialización y distribución de beneficios	9
V-A8.	Condiciones agroclimáticas y recursos	9
V-A9.	Tipología de productores	10
V-A10.	Problemas principales de la cadena	10
V-A11.	Fortalezas y debilidades de la cadena	10
V-A12.	Producción en la Provincia del Guayas	10
V-A13.	Factores que limitan la productividad	10
V-A14.	Análisis estadístico de la producción	10
V-A15.	Producción en Manabí	11
V-A16.	Factores que limitan la producción	12
V-A17.	Problemática del valor agregado	13
V-A18.	Potencial productivo y oportunidades	13
V-B.	Inteligencia Artificial y Visión Artificial Aplicadas a la Agricultura y ganadería	13
V-B1.	Visión artificial como estrategia de transformación agrícola	13
V-B2.	Rendimiento de la visión artificial en la agricultura	14
V-B3.	Visión artificial en la ganadería	15
V-B4.	Rendimiento de la visión artificial en la ganadería	17
V-C.	Algoritmos y modelos de detección de objetos en visión artificial	19
V-C1.	Comparación y aplicaciones	20
V-D.	Redes Neuronales Convolucionales (CNNs) para el Análisis de Imágenes	20
V-D1.	Estructura básica de una CNN	21
V-D2.	Aplicación de CNN en la Clasificación de Maduración del Cacao	22
V-D3.	Retos y Consideraciones	22
V-E.	Google Colab	22
V-F.	Labelling	23
V-G.	Visual Studio 2019	24
V-H.	Redes Pre-entrenadas y Transfer Learning	24

VI. Marco metodológico	25
VI-A. Adquisición de la imagen	25
VI-B. Clases del dataset elegido	25
VI-C. Preprocesamiento	25
VI-D. Segmentación	25
VI-E. Extracción de características	25
VI-F. Etiquetación	26
VI-G. Clasificación	26
VI-G1. Parámetros del modelo CNN	26
VI-H. Captura de imágenes y visualización de resultados	26
VI-I. Importación de librerías y montaje del entorno	27
VI-J. Definición de rutas y conteo de imágenes	28
VI-K. Recolección de rutas de imágenes	29
VI-L. Generadores de datos para entrenamiento y validación	30
VI-M. Definición y compilación del modelo de red neuronal convolucional (CNN)	31
VI-N. Entrenamiento del modelo	33
VI-Ñ. Visualización del Rendimiento del Modelo	35
VI-O. Evaluación del Modelo y Predicción en Tiempo Real	37
VII. Resultados	40
VII-A. Funcionamiento del sistema	40
VII-B. Evaluación de el sistema de predicción	40
VII-B1. Resultados de el cacao inmaduro	40
VII-B2. Resultados de el cacao maduro	41
VII-B3. Resultados de el cacao sobre maduro	42
VIII. Cronograma	45
IX. Presupuesto	46
X. Conclusiones	47
XI. Recomendaciones	48
Referencias	49
Anexo A: CÓDIGO USADO EN GOOGLE COLAB PARA LA REALIZACIÓN DEL MODELO DEL SISTEMA DE RECONOCIMIENTO DE VISIÓN ARTIFICIAL	51
Anexo B: CÓDIGO DE PYTHON USADO EN VISUAL STUDIO CODE PARA EL RECONOCIMIENTO DE VISIÓN ARTIFICIAL	54
Anexo C: CÓDIGO HTML PARA LA PAGINA DE INICIO DE LA PAGINA WEB	56
Anexo D: CÓDIGO HTML PARA LA PAGINA DE PREDICCIÓN DE LA PAGINA WEB	58
Anexo E: EJEMPLO DE DATASETS USADOS PARA EL ENTRENAMIENTO DEL MODELO TANTO INMADURO, MADURO Y SOBRE MADURO RESPECTIVAMENTE	60

ÍNDICE DE FIGURAS

1.	Tipos de cacao [16]	5
2.	Planta de Cacao [16]	7
3.	Diferentes mazorcas de cacao en su maduración [18]	7
4.	Cacao fino de aroma en la provincia de los ríos [21]	8
5.	Agricultores y la materia prima [22]	9
6.	Cacao de la provincia del Guayas [23]	11
7.	Cacao de la provincia de Manabí [23]	12
8.	Visión artificial en producción de tomates [26]	14
9.	Dron como herramienta para la detección de un campo agrícola [27]	15
10.	Estimación del rendimiento de la visión artificial en la agricultura [28]	16
11.	Visión artificial en la ganadería [29]	16
12.	Visión artificial aplicada en bovinos Holstein Friesian [30]	17
13.	Procesamiento para entrenamiento y formación de base de datos [30]	18
14.	Estimación de numero de aves mediante procesamiento de imágenes [31]	18
15.	Modelo desarrollado por microsoft Faster R-CNN [33]	19
16.	Modelo YOLO enfocado en la detección en tiempo real [33]	20
17.	Estructura general del modelo SSD (Single Shot Detector) utilizado para la detección de objetos en una sola pasada. [33]	20
18.	Estructura de una CNN [35]	22
19.	Google Colab [36]	23
20.	labelling [37]	23
21.	Visual Studio 2019 [38]	24
22.	Arquitectura del Transfer Learning [40]	24
23.	Procesamiento de imagenes con python. Autoría propia	27
24.	Importación de librerías. Autoría propia	28
25.	Definición de rutas y conteo de imágenes. Autoría propia	28
26.	Definición de rutas. Autoría propia	28
27.	conteo de imágenes. Autoría propia	29
28.	Recolección de rutas de imágenes. Autoría propia	29
29.	Entrenamiento y validación del modelo. Autoría propia	30
30.	Entrenamiento y validación del modelo	30
31.	Entrenamiento y validación del modelo. Autoría propia	31
32.	compilación del modelo CNN. Autoría propia	31
33.	compilación del modelo CNN. Autoría propia	32
34.	compilación del modelo CNN. Autoría propia	33
35.	Entrenamiento del modelo. Autoría propia	33
36.	Entrenamiento del modelo en 10 épocas. Autoría propia	34
37.	Visualización del Rendimiento. Autoría propia	35
38.	Precisión en la predicción. Autoría propia	35
39.	Precisión del modelo. Autoría propia	36
40.	Pérdidas del modelo. Autoría propia	37
41.	Evaluación del Modelo. Autoría propia	38
42.	Predicción sobre imagen procesada. Autoría propia	39
43.	Guardado del modelo. Autoría propia	39
44.	Imagen procesada y evaluada. Autoría propia	40
45.	Evaluación cacao inmaduro. Autoría propia	40
46.	Aplicación de visión artificial para el cacao inmaduro.	41
47.	Evaluación cacao maduro. Autoría propia	41
48.	Aplicación de visión artificial para el cacao maduro. Autoría propia	42

49.	Evaluación cacao sobre maduro. Autoría propia	42
50.	Aplicación de visión artificial para el cacao sobre maduro. Autoría propia	43
51.	Sistema de visión artificial en la pagina web. Autoría propia	43

ÍNDICE DE TABLAS

I.	Estados de madurez del cacao	6
II.	Cantidad de imágenes por clase de maduración del cacao. Autoría Propia.	25
III.	Arquitectura de la red neuronal convolucional propuesta	26
IV.	Cronograma	45
V.	Presupuesto	46

I. INTRODUCCIÓN

El cacao fino de aroma representa uno de los principales productos agrícolas de exportación del Ecuador, consolidando al país como uno de los mayores productores a nivel mundial. Esta variedad, valorada por su sabor, calidad y propiedades organolépticas, tiene una creciente demanda internacional, lo cual genera oportunidades significativas para los productores nacionales. Sin embargo, uno de los principales retos en su cadena de valor es la correcta identificación del momento óptimo de cosecha, ya que la maduración de las mazorcas influye directamente en la calidad final del grano y, por tanto, en su valor comercial.

Tradicionalmente, el grado de maduración de las mazorcas de cacao ha sido evaluado de forma visual y manual, dependiendo del conocimiento empírico de los agricultores. Este método, aunque funcional, está sujeto a errores humanos, condiciones ambientales y diferencias entre variedades. La recolección prematura o tardía puede afectar la fermentación, generar pérdidas económicas y comprometer el cumplimiento de estándares de exportación. En este contexto, la incorporación de herramientas tecnológicas que faciliten y automaticen este proceso se vuelve no solo deseable, sino necesaria, especialmente para los pequeños productores que representan más del 90 % del sector cacaotero ecuatoriano.

Ante esta problemática, el presente proyecto propone el desarrollo de un sistema de detección automática del grado de maduración del cacao mediante el uso de un aplicativo móvil basado en visión artificial. Utilizando modelos de inteligencia artificial como YOLOv5, entrenados con imágenes reales de mazorcas en distintos estados de maduración, se busca crear una herramienta portátil, eficiente y de fácil uso. Esta aplicación permitirá al usuario capturar imágenes con su teléfono celular, procesarlas localmente sin necesidad de conexión a internet y recibir información precisa en tiempo real sobre el estado de madurez de los frutos. De este modo, se contribuye a una agricultura de precisión, optimizando los procesos de cosecha y fortaleciendo la productividad y sostenibilidad del cultivo del cacao en el Ecuador.

II. PROBLEMA

La producción de cacao en Ecuador es una de las principales actividades agrícolas del país, con más de 600,000 hectáreas cultivadas y más de 400,000 familias involucradas en su cultivo. Sin embargo, uno de los principales desafíos que enfrentan los productores es la determinación del punto óptimo de maduración de las mazorcas de cacao para su recolección. Esta etapa es crítica, ya que influye directamente en la calidad organoléptica, el rendimiento del grano, y el valor económico del producto final [1][2].

Actualmente, la madurez de las mazorcas es evaluada mayormente a través de observación visual por parte del agricultor, basándose en el cambio de color de la cáscara. No obstante, este método presenta una alta subjetividad, está condicionado por la experiencia del recolector y puede verse afectado por factores como la luz solar, enfermedades superficiales, o la variabilidad genética entre variedades [3][4]. Este tipo de error puede llevar a la cosecha prematura o tardía de las mazorcas, provocando pérdidas económicas importantes y disminuyendo la calidad del producto final.

Además, la dispersión geográfica de los cultivos y las condiciones climáticas cambiantes hacen difícil implementar controles de madurez eficientes en plantaciones medianas o grandes, lo que incrementa la necesidad de un sistema automatizado, accesible y portátil que asista al productor durante la recolección [5]. Pese a los avances en agricultura de precisión y el uso de sensores avanzados, existe un vacío tecnológico respecto a herramientas de bajo costo que sean aplicables al entorno rural ecuatoriano, en especial en zonas con acceso limitado a infraestructura tecnológica [6].

III. JUSTIFICACIÓN

La implementación de un sistema de monitoreo visual automatizado para determinar el grado de madurez del cacao tiene múltiples beneficios técnicos, sociales y económicos. En primer lugar, la objetividad que brinda la visión artificial puede ayudar a reducir significativamente el error humano en la recolección, optimizando la calidad del grano fermentado y del chocolate final [7][8]. De esta manera, se eleva el estándar de producción, permitiendo a pequeños productores cumplir con exigencias de mercados internacionales que demandan trazabilidad y consistencia en la calidad del producto [9].

Desde el punto de vista técnico, los avances recientes en modelos de clasificación de imágenes y el procesamiento de color han demostrado ser altamente eficientes incluso en ambientes no controlados, utilizando únicamente cámaras RGB y técnicas de aprendizaje supervisado o no supervisado [10]. Esto permite que el sistema pueda funcionar en campo sin necesidad de cámaras de alto valor o infraestructura compleja, haciéndolo accesible a comunidades agrícolas con recursos limitados [11].

Además, al tratarse de un cultivo estratégico para la economía ecuatoriana, cualquier mejora en la eficiencia de la producción de cacao puede tener un impacto significativo en el desarrollo rural, la sostenibilidad ambiental y la competitividad internacional del país [12]. Un sistema de este tipo también puede recopilar datos históricos de cada plantación, ayudando en procesos de planificación, predicción de rendimientos y análisis de fenología vegetal [13].

En un contexto más amplio, este tipo de solución tecnológica fomenta la digitalización de la agricultura, promoviendo una transición hacia la agricultura inteligente y sostenible, alineada con los objetivos de desarrollo sostenible (ODS) planteados por organismos internacionales como la FAO [14][15].

IV. OBJETIVOS

IV-A. Objetivo general

Desarrollar un sistema de visión artificial basado en redes neuronales convolucionales implementado en un aplicativo móvil que permita la detección del grado de maduración del cacao mediante análisis de patrones de las mazorcas.

IV-B. Objetivos específicos

1. Diseñar un algoritmo de procesamiento de imágenes para la identificación y clasificación considerando el tamaño y color de las mazorcas de cacao según su etapa de maduración.
2. Implementar una aplicación web funcional que integre el modelo de clasificación y permita la detección de mazorcas de manera eficiente.
3. Validar la exactitud del sistema mediante pruebas de detección de grados de maduración de las mazorcas de cacao en campo.

V. MARCO TEÓRICO

V-A. El cultivo del cacao en Ecuador

El cacao ecuatoriano es reconocido mundialmente por ser cacao fino y de aroma, especialmente la variedad “Nacional” o “Arriba”, la cual constituye aproximadamente el 60 % de la producción mundial de cacao aromático. Ecuador posee alrededor de 475,000 hectáreas cultivadas de cacao, distribuidas entre huertas viejas (42 %) con genotipos nacionales y cruza, híbridos y clones recomendados, y cerca de 150,000 hectáreas con el clon CCN-51[1]. CCN-51: Un clon más moderno utilizado por su mayor rendimiento productivo, aunque menos apreciado por la calidad del aroma en comparación con el Nacional. Representa cerca del 28 % de la superficie cultivada en Ecuador.

V-A1. Características y variedades:

■ Variedad Nacional:

Es la más emblemática, con un perfil de sabor floral, frutal y aromático, incluyendo notas tropicales como piña y mango. La mazorca puede ser de forma amelonada o cundeamor, con color maduro que varía desde amarillo hasta amarillo cenizo o café. Las semillas suelen ser grandes, de forma redonda achatada o cilíndrica, con color púrpura oscuro o claro. Estas plantas son en su mayoría auto-compatibles y presentan diversas resistencias moderadas a enfermedades como la monilla.

■ Criollo y Forastero:

Si bien están presentes en Ecuador, la mayoría del cacao comercial es Nacional o híbridos con características de aroma fina o clones como el CCN-51, sus granos son delgados, de color más claro y menos amargo que otras variedades como el cacao forastero o trinitario.

■ CCN-51:

Un clon más moderno utilizado por su mayor rendimiento productivo, aunque menos apreciado por la calidad del aroma en comparación con el Nacional. Representa cerca del 28 % de la superficie cultivada en Ecuador como se observa en la imagen 1.



Figura 1. Tipos de cacao [16]

V-A2. *Estado de madurez del cacao:*

- La mazorca cambia de color desde un verde inmaduro a tonos como amarillo, amarillo cenizo o café en el estado maduro. La forma de la mazorca puede presentar protuberancias específicas como barriga y punta tipo pico de loro.
- La flor del cacao tiene pigmentación en los estambres que puede oscilar entre rojo pálido a rojo oscuro, dependiendo del clon.
- El índice de mazorca (número de mazorcas necesarias para producir 1 kg de cacao fermentado y seco) varía típicamente entre 17 y 32 mazorcas según la variedad y clon.

Tabla I
ESTADOS DE MADUREZ DEL CACAO

Estado de Madurez	Características Visuales de la Mazorca	Coloración de la Mazorca y Surcos	Estado de las Semillas
Estado 1	Caballetes color violeta oscuro; surcos verdes	Mazorca verde, sin madurez aparente	Semillas poco desarrolladas
Estado 2	Surcos con tonalidades amarillas en algunos puntos	Mazorca verde con surcos amarillentos	Semillas iniciando desarrollo
Estado 3	Tonalidad violeta menos intensa, amarillo alrededor	Parte amarilla creciendo alrededor de la mazorca	Semillas en mayor desarrollo

El cacao (*Theobroma cacao*) es un cultivo de alto valor económico y social en países productores como Ecuador, Colombia, Perú y otros de América Latina, África y Asia. El fruto de cacao pasa por diversas etapas de maduración que se reflejan visualmente en su color, textura y forma. Generalmente, los estados de maduración se clasifican en:

- Verde: fruto inmaduro, sin desarrollo adecuado de los granos.
- Amarillo o intermedio: etapa de transición, no siempre adecuada para cosecha.
- Marrón o maduro: punto óptimo de cosecha, donde el grano ha alcanzado su desarrollo y contenido de azúcares.
- Pasado o sobre maduro: puede presentar fermentación interna, riesgo de pérdida de calidad.

Identificar visualmente estas etapas de maduración puede resultar subjetivo y depender de la experiencia del agricultor. De ahí la necesidad de implementar sistemas automáticos que reduzcan errores y estandaricen el proceso de clasificación.

Ecuador se erige como un país de tradición cacaotera, cuyo cultivo no solo sustenta economías locales y nacionales, sino que también forma parte intrínseca de su patrimonio cultural e histórico. La calidad y el aroma distintivo de variedades como el “Cacao Arriba” han posicionado al país en mercados internacionales exigentes. Dentro de este contexto comprender las diversas variedades de cacao cultivadas, como el tradicional “Nacional” con sus notas florales y frutales, y las variedades más productivas como el CCN-51, resulta importante, ya que sus características fenotípicas pueden variar durante la maduración [17].

El proceso de maduración del fruto del cacao es un fenómeno complejo que involucra cambios bioquímicos y físicos evidentes. Visualmente, se transita por diversas etapas caracterizadas por alteraciones en el color de la

mazorca (desde verde intenso hasta amarillo, naranja o rojo, dependiendo de la variedad), modificaciones en la forma y el tamaño, y variaciones en la textura de la cáscara. Identificar estos atributos visuales en cada etapa (Inmaduro, maduro, sobremaduro) es fundamental para definir las clases que el sistema de visión artificial deberá distinguir [18]. Los métodos tradicionales de evaluación, basados principalmente en la experiencia del productor a través de la inspección visual y el tacto, adolecen de subjetividad y pueden llevar a inconsistencias en la determinación del punto óptimo de cosecha. Además, algunos métodos implican la apertura de frutos, lo que impide su posterior comercialización. Esta necesidad de una evaluación más objetiva, consistente y no destructiva motiva la exploración de tecnologías como la visión artificial.



Figura 2. Planta de Cacao [16]

La mazorca de cacao inicia su maduración de forma gradual una vez completada la polinización y la formación del fruto. Durante esta etapa, el fruto aumenta de tamaño y se va llenando de semillas de cacao. Simultáneamente, la pulpa que rodea las semillas madura hasta adquirir un sabor dulce característico, un factor clave para el éxito de la fermentación posterior. En conjunto, estos cambios internos (crecimiento de las semillas y desarrollo de la pulpa) constituyen la fase fundamental del proceso de maduración del cacao [18]

A medida que la mazorca progresa en su desarrollo, también experimenta transformaciones visibles en su aspecto externo. En particular, la cáscara del fruto cambia gradualmente de un tono verde inicial hacia colores más amarillos, rojos o anaranjados, dependiendo de la variedad de cacao cultivada. Paralelamente, la superficie de la mazorca se torna más suave al tacto. Estos cambios en el color y la textura de la cáscara sirven como indicadores fiables de que el fruto está alcanzando la madurez, ya que sólo en ese punto la mazorca presenta las condiciones óptimas de desarrollo interno y de calidad de la semilla.



Figura 3. Diferentes mazorcas de cacao en su maduración [18]

V-A3. *Producción y rendimiento:* La introducción del cacao en Ecuador se remonta a mediados del siglo XVI, cuando comenzó a cultivarse en las riberas de los ríos de Guayaquil. A lo largo de los siglos, las provincias de Guayas, Los Ríos y El Oro se consolidaron como los principales centros productores. Durante el boom cacaotero de 1860, el cultivo llegó a constituir alrededor del 70 % de las exportaciones nacionales, marcando una época de prosperidad económica vinculada directamente a la producción de esta provincia. Aunque con el tiempo surgieron crisis vinculadas a plagas, caída de precios internacionales y cambios en la estructura agraria, Guayas ha mantenido una participación significativa dentro del mapa cacaotero del país. [19]

En Ecuador, la producción y el rendimiento del cacao dependen en gran medida de las prácticas agrícolas aplicadas. Bajo métodos tradicionales, el rendimiento promedio ronda los 400 kg por hectárea, mientras que la implementación de técnicas modernas, como el riego de precisión, permite alcanzar entre 2,5 y 3 toneladas por hectárea, de acuerdo con datos de Netafim. El país ocupa el tercer lugar a nivel mundial en producción de cacao y es líder en cacao fino de aroma, con una producción anual cercana a las 400 mil toneladas. Las provincias de Los Ríos, Guayas y Manabí destacan como las principales zonas productoras.[20]

V-A4. *Producción en la Provincia de los Ríos:* En la provincia de Los Ríos, en Ecuador, constituye una de las principales zonas productoras de cacao del país, reconocido mundialmente por la calidad de su Cacao Fino de Aroma. El estudio analiza de manera detallada la estructura, funcionamiento y problemas de la cadena productiva del cacao en esta región, poniendo énfasis en los actores involucrados, la dinámica de producción, las condiciones ambientales y los márgenes de rentabilidad[2].



Figura 4. Cacao fino de aroma en la provincia de los ríos [21]

V-A5. *Producción y características de los agricultores:* La mayor parte de la producción proviene de pequeños y medianos agricultores; 70 % y 20 % respectivamente, mientras que los grandes productores representan apenas un 10 %. Los pequeños agricultores presentan limitaciones significativas:

- Poca mecanización en sus labores agrícolas.
- Uso intensivo, pero ineficiente, de fertilizantes.
- Empleo de semillas disponibles, no siempre de óptima calidad.
- Baja densidad de siembra y ausencia de buenas prácticas agrícolas.
- Rendimientos reducidos, en promedio de 0,67 toneladas por hectárea, lo que es bajo frente al potencial productivo.

Este contexto limita la rentabilidad de los productores, lo que en muchos casos los empuja a diversificar hacia otros cultivos, generando un efecto de arrastre que también repercute en medianos y grandes agricultores.

V-A6. *Evolución de la producción (2015-2018)*: El análisis muestra un crecimiento progresivo en la producción de cacao en Los Ríos. En 2015 se registraron 36.185 toneladas, mientras que en 2018 se alcanzó el pico máximo con 70.374 toneladas. Sin embargo, la superficie sembrada varió: en 2017 se registró el mayor nivel (130.585 ha), cayendo en 2018 a 105.503 ha. Esto indica que la productividad por hectárea mejoró, pero persisten limitaciones técnicas que frenan la competitividad.

V-A7. *Comercialización y distribución de beneficios*: Los pequeños agricultores comercializan el cacao en baba o semi seco principalmente a través de intermediarios y mayoristas[22]. Este esquema hace que los beneficios por calidad (premio de mercado) sean captados por exportadores y mayoristas, y no por los agricultores. Un análisis de márgenes realizado en 2018-2019 revela la inequidad en la distribución de las ganancias:

- Productores: 7 % de utilidad.
- Minoristas: 10 %.
- Organizaciones de base: 8
- Exportadores: 16

Esto refleja que, pese a ser los responsables del mayor esfuerzo productivo y de enfrentar riesgos climáticos y de mercado, los agricultores son quienes menos beneficio obtienen.



Figura 5. Agricultores y la materia prima [22]

V-A8. *Condiciones agroclimáticas y recursos*: El estudio resalta que Los Ríos cuenta con condiciones naturales favorables:

- Precipitación entre 1.600 y 2.500 mm anuales, adecuada para el cultivo.
- Temperaturas medias de 25°C, óptimas para floración y fructificación.
- Suelos fértiles, aluviales y profundos, con buena capacidad de retención de agua.

No obstante, también existen debilidades: deficiente manejo del agua por falta de infraestructura de riego y drenaje, uso inadecuado de fertilizantes y pesticidas, ausencia de análisis de suelo, baja incorporación de materia orgánica y poca innovación tecnológica.

V-A9. *Tipología de productores:* El estudio diferencia tres perfiles de agricultores:

- Tradicionales: baja adopción tecnológica, poca fertilización, semillas de baja calidad, rendimientos de 0,5 a 0,6 t/ha y escasa participación en organizaciones.
- Emprendedores o progresistas: poseen más recursos, siembran clones (CCN51 principalmente), manejan mejor los fertilizantes y controlan plagas; alcanzan rendimientos de 0,6 a 0,8 t/ha.
- Medianos y grandes agricultores: con acceso a mecanización, semillas mejoradas y buenas prácticas, superan las 0,8 t/ha y tienen mayor libertad de elegir canales de comercialización.

V-A10. *Problemas principales de la cadena:* El análisis identifica varios cuellos de botella que frenan la competitividad:

- Agua: mala distribución de riego, priorización hacia otros cultivos como arroz y caña, corrupción en el acceso al recurso.
- Suelos: compactación por mecanización deficiente, falta de análisis de fertilidad y baja aplicación de materia orgánica.
- Semillas: limitada disponibilidad de material vegetativo de alto rendimiento.
- Cosecha y post-cosecha: escasez de infraestructura de almacenamiento y dependencia de prestamistas que condicionan la venta.
- Comercialización: fuerte presencia de intermediarios que reducen el precio pagado al productor.

V-A11. *Fortalezas y debilidades de la cadena:* **Fortalezas:** clima favorable, disponibilidad de mano de obra, disposición de los agricultores a pagar por asistencia técnica, potencial de expansión del cultivo. **Debilidades:** deficiencias técnicas en producción, baja asociatividad, escasa innovación, problemas de financiamiento, concentración de beneficios en intermediarios y exportadores.

V-A12. *Producción en la Provincia del Guayas:* Aunque Guayas sigue siendo una zona importante de producción, la rentabilidad ha disminuido en los últimos años por los bajos niveles de productividad.

Según el Ministerio de Agricultura (MAG, 2018), en 2016 la superficie sembrada de cacao en Guayas fue de 101.724 hectáreas, un 33 % menos respecto al 2004. A pesar de esta reducción, el rendimiento por hectárea creció ligeramente (+0,14 toneladas por ha), aunque aún se mantiene por debajo de los niveles alcanzados en África. El cultivo de cacao en Guayas constituye el sustento económico de numerosas familias rurales en cantones como Milagro, Yaguachi, El Empalme, Juján y Naranjal. La mayoría de estas explotaciones son pequeñas y medianas unidades productivas: el 88 % son fincas agropecuarias de 20 a 50 hectáreas, que abarcan el 73,4 % del área cultivada. [23]

V-A13. *Factores que limitan la productividad:* La producción se ve afectada principalmente por:

- Edad avanzada de los árboles.
- Suelos poco fertilizados.
- Deficiente control de plagas y enfermedades.

Estas condiciones explican los bajos niveles de producción y la necesidad de capacitaciones, transferencia tecnológica y manejo agronómico mejorado para elevar la productividad.

V-A14. *Análisis estadístico de la producción:* En una evaluación la producción en distintos cantones de Guayas. Los resultados mostraron que:



Figura 6. Cacao de la provincia del Guayas [23]

- No existen diferencias significativas de producción entre cantones, lo que significa que las dificultades productivas son generalizadas en toda la provincia.
- Dentro de cada cantón sí se observa una alta variabilidad entre productores, lo que refleja desigualdades en acceso a recursos, prácticas agrícolas y disponibilidad de mano de obra.

Las variables que más influyen en la producción son:

- Número de hectáreas cultivadas.
- Disponibilidad y organización de la mano de obra.
- Acceso a herramientas y equipos.

Esto sugiere que la productividad no depende solo de factores naturales, sino también del nivel de recursos y gestión que cada productor pueda aplicar en su finca.

V-A15. Producción en Manabí: La provincia de Manabí es reconocida como una de las principales zonas cacaoteras del Ecuador. Desde tiempos coloniales, este cultivo ha formado parte de la identidad productiva y cultural de la región, generando ingresos, empleo y dinamizando la economía rural. Sin embargo, a pesar de su importancia histórica, la producción de cacao enfrenta actualmente un conjunto de dificultades que limitan su productividad y competitividad frente a los mercados nacionales e internacionales.

El cacao en Manabí no es solo un producto de exportación, sino también un sustento económico y social para miles de familias campesinas. La mayoría de los productores son pequeños y medianos agricultores que dependen directamente de este cultivo como su principal fuente de ingresos.



Figura 7. Cacao de la provincia de Manabí [23]

El cacao ecuatoriano, y en particular el de Manabí, tiene el reconocimiento internacional de ser un cacao fino de aroma, con características de sabor y calidad muy valoradas en la industria chocolatera mundial. Sin embargo, esta ventaja comparativa no se traduce de manera directa en mejores ingresos para los agricultores manabitas, ya que la producción se ve frenada por factores estructurales.

La producción de cacao en Manabí presenta un panorama mixto: Por un lado, existe un alto potencial agrícola gracias a sus condiciones agroclimáticas favorables, suelos fértiles y tradición agrícola. Por otro lado, los niveles de productividad son bajos, lo que genera una brecha entre el potencial y la realidad.

Gran parte de las fincas utilizan métodos de cultivo tradicionales, con baja mecanización y limitada asistencia técnica. Esto provoca rendimientos reducidos, incluso en comparación con otras provincias ecuatorianas y con países productores de África Occidental, que dominan el mercado mundial. [24]

V-A16. Factores que limitan la producción: La baja productividad del cacao manabita responde a una serie de problemas estructurales:

1. Muchos cultivos son antiguos y no han sido renovados, lo que disminuye el rendimiento de las hectáreas.
2. Los productores carecen de acceso a semillas certificadas o a clones de alta productividad.
3. Escasa innovación tecnológica: el cultivo se mantiene con prácticas rudimentarias y poco tecnificadas.
4. Plagas y enfermedades: como la moniliasis o la escoba de bruja, que afectan la calidad y la cantidad de la producción.
5. Problemas ambientales: la provincia es vulnerable a fenómenos naturales como sequías, inundaciones o el impacto de El Niño, que afectan las cosechas.
6. Limitado acceso a financiamiento: los pequeños productores no suelen contar con créditos agrícolas que les permitan invertir en fertilización, riego o tecnificación.
7. Débil apoyo institucional: la asistencia técnica y el acompañamiento estatal son escasos, lo que genera una brecha de conocimiento en los agricultores.

V-A17. *Problemática del valor agregado:* Uno de los grandes problemas de la producción en Manabí es que el cacao se comercializa casi exclusivamente como materia prima. Esto significa que los productores venden el grano en estado natural (en baba o seco) a intermediarios o compradores mayoristas, sin procesarlo.

Al no industrializarse en la misma provincia, los agricultores pierden la oportunidad de captar mayores márgenes de ganancia que podrían generarse en la elaboración de chocolates, manteca o licor de cacao. La cadena de valor se concentra en los intermediarios y exportadores, dejando a los agricultores con ingresos reducidos y poco competitivos frente al esfuerzo que representa producir. [25]

V-A18. *Potencial productivo y oportunidades:* A pesar de los problemas, la provincia de Manabí posee un enorme potencial para convertirse en un referente en la producción de cacao de calidad:

- Sus condiciones agroclimáticas son ideales para el cultivo.
- Existe una demanda mundial insatisfecha de cacao fino y de aroma, lo que abre oportunidades en mercados especializados.
- La tradición agrícola de la provincia asegura un conocimiento acumulado que puede fortalecerse con asistencia técnica y acceso a tecnología.
- La asociatividad de los productores aparece como una estrategia clave: mediante cooperativas o asociaciones, se podría negociar mejores precios, acceder a créditos y reducir la dependencia de intermediarios.

V-B. *Inteligencia Artificial y Visión Artificial Aplicadas a la Agricultura y ganadería*

La Inteligencia Artificial (IA) tiene un amplio espectro de técnicas computacionales con el propósito de simular el cognitivo de un humano, teniendo la capacidad para resolver problemas, o aprender como un humano lo haría. Dentro de la IA, el Aprendizaje Automático (Machine Learning) permite que las computadoras sean capaces de aprender datos, mediante el desarrollo de algoritmos, quitando la necesidad de ser programado. La Visión Artificial (Computer Vision) es un campo específico que busca dotar a las máquinas con la capacidad de ver e interpretar el mundo visual a través del procesamiento y análisis de imágenes digitales[21].

V-B1. *Visión artificial como estrategia de transformación agrícola:* La implementación de la visión artificial en las operaciones agrícolas requiere un enfoque estratégico y progresivo orientado a la obtención de resultados medibles. Para ello, es fundamental que los agricultores y las empresas del sector alineen estas tecnologías con objetivos concretos, tales como incrementar el rendimiento de los cultivos, reducir los costos de los insumos o fortalecer las prácticas de sostenibilidad[22].

El proceso inicia con la identificación de aquellas áreas clave en las que la automatización y el análisis de datos aportan un mayor valor, como la monitorización de la salud del ganado, la gestión del crecimiento de los cultivos o la optimización del uso de los recursos disponibles[26].

Posteriormente, la selección de las herramientas adecuadas como drones, dispositivos IoT o sistemas avanzados de captura y análisis de imágenes resultan esencial para garantizar una integración fluida con los flujos de trabajo ya establecidos[26].

La compatibilidad con sistemas de gestión agrícola y plataformas en la nube, además, contribuye a reforzar la escalabilidad y el control de la información, elementos indispensables para una gobernanza tecnológica sólida[21].



Figura 8. Visión artificial en producción de tomates [26]

La adopción de la visión artificial permite, en consecuencia, aprovechar el potencial no explotado de las operaciones agrícolas, facilitando la creación de prácticas más escalables, rentables y sostenibles. Sus beneficios se traducen en mejoras tangibles de eficiencia y productividad, además de un impacto directo en la rentabilidad[21].

De esta manera, la visión artificial se posiciona como una herramienta transformadora que impulsa soluciones innovadoras, optimiza la asignación de recursos y contribuye al fortalecimiento de la sostenibilidad en el sector agrícola.

V-B2. Rendimiento de la visión artificial en la agricultura: La estimación del rendimiento constituye una práctica esencial en la etapa de precosecha dentro de la mayoría de las empresas agrícolas de gran escala. Este proceso permite tomar decisiones estratégicas respecto a la asignación de recursos logísticos, tales como la planificación del transporte, la distribución de la mano de obra y la gestión de insumos[26]. Una sobreestimación de la producción puede generar un exceso de costos operativos que afectan directamente la rentabilidad, mientras que una subestimación puede conllevar desperdicio de cultivos y gastos adicionales en almacenamiento o comercialización. Por ello, contar con estimaciones precisas resulta fundamental para garantizar una gestión eficiente y sostenible de la producción agrícola[26].

Además de su relevancia logística, la predicción del rendimiento también se convierte en una herramienta clave para la optimización de las prácticas de cultivo. Disponer de información anticipada sobre el volumen de producción esperado permite ajustar estrategias de fertilización, riego o control fitosanitario, reduciendo así los riesgos asociados a enfermedades, plagas o deficiencias nutricionales de los cultivos.

En este contexto, los avances tecnológicos, particularmente en el campo de la visión artificial y el aprendizaje profundo, han impulsado el desarrollo de algoritmos basados en redes neuronales convolucionales profundas (CNNs) que facilitan una predicción precisa del rendimiento[27].

Estas técnicas permiten el conteo automático de frutas y hortalizas a partir de imágenes capturadas en campo,



Figura 9. Dron como herramienta para la detección de un campo agrícola [27]

incluso en condiciones complejas que incluyen oclusiones por hojas o ramas, variaciones en la iluminación y diferencias en el tamaño de los objetos. De esta manera, los sistemas modernos de visión artificial han demostrado una capacidad robusta para superar las limitaciones que tradicionalmente dificultaban las mediciones manuales[27].

El método tradicional de estimación, basado en el conteo manual de productos como frutas o verduras, implica un trabajo intensivo en mano de obra y resulta costoso, además de estar sujeto a errores humanos que comprometen la exactitud de los resultados.

Frente a ello, la aplicación de visión artificial ofrece una alternativa más eficiente, automatizada y escalable, que no solo reduce significativamente los costos, sino que también incrementa la precisión y rapidez del proceso[21].

V-B3. Visión artificial en la ganadería: La capacidad de la visión artificial para procesar grandes volúmenes de datos visuales de manera rápida, objetiva y precisa la convierte en una tecnología prometedora para optimizar diferentes tareas dentro de la ganadería[29].

Entre sus usos más relevantes se encuentran el control del comportamiento animal, la detección de irregularidades en la alimentación o en el consumo de agua, la supervisión del desplazamiento en corrales y pastizales, así como el reconocimiento individual de cada animal mediante patrones visuales[21].

Estas aplicaciones permiten no solo incrementar la eficiencia en el manejo del ganado, sino también fortalecer las estrategias de bienestar animal y reducir los costos asociados a la atención veterinaria preventiva, podemos hablar de algunos beneficios de aplicar este sistema en la ganadería actualmente como puede ser:

- **Monitoreo mejorado del ganado:** La utilización de imágenes de alta resolución posibilita una evaluación precisa del estado de salud de los animales, facilitando la detección temprana de factores de estrés ocasionados por enfermedades, parásitos o deficiencias nutricionales. Este diagnóstico anticipado reduce de manera significativa las pérdidas productivas.
- **Manejo eficiente de enfermedades y parásitos:** Los sistemas automatizados de detección permiten identificar animales o áreas específicas que requieren atención, orientando la aplicación focalizada de tratamientos veterinarios y antiparasitarios. Este enfoque no solo reduce el uso innecesario de medicamentos, sino que

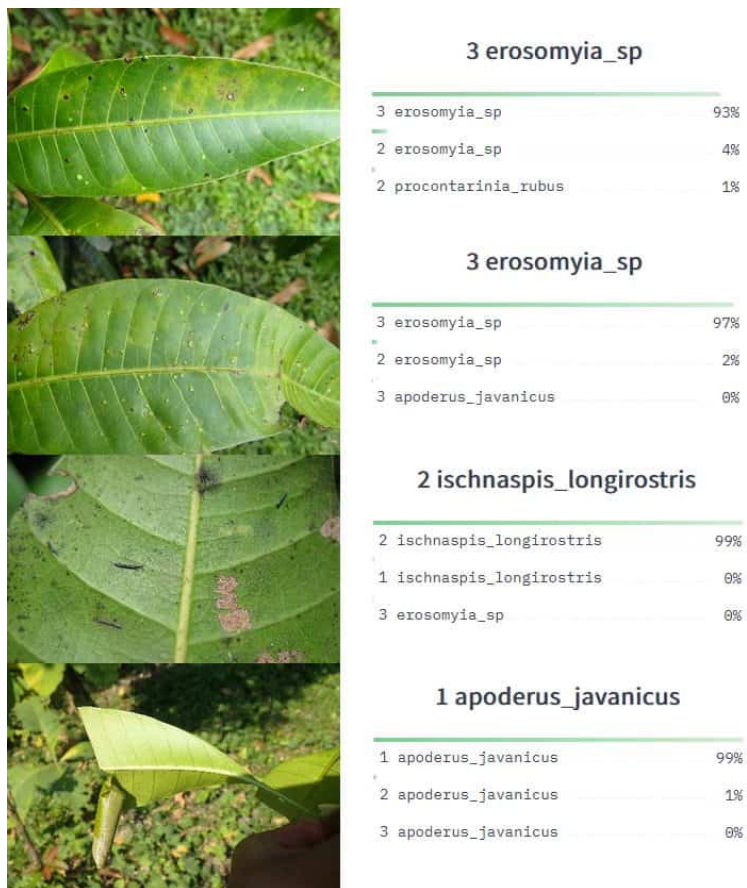


Figura 10. Estimación del rendimiento de la visión artificial en la agricultura [28]

también disminuye los costos operativos y minimiza el impacto sobre la salud del rebaño.

- Ganadería de precisión: La visión artificial optimiza la gestión de recursos al garantizar que el alimento, el agua y los tratamientos veterinarios se suministren únicamente cuando y donde sean necesarios. Esto permite minimizar el desperdicio, reducir los costos de producción y maximizar la eficiencia y rentabilidad del manejo del ganado.

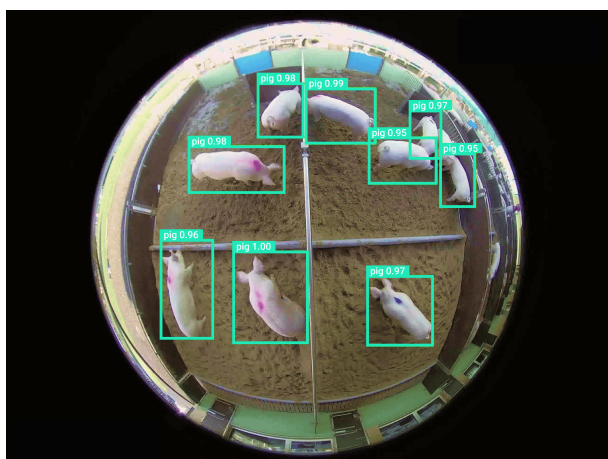


Figura 11. Visión artificial en la ganadería [29]

V-B4. *Rendimiento de la visión artificial en la ganadería:* En el año 2021 se presentó un estudio en el que se desarrolló un sistema de visión por computadora basado en técnicas de aprendizaje profundo, con el propósito de reconocer en tiempo real vacas individuales, identificar sus posiciones, acciones y movimientos, así como registrar los historiales temporales de cada animal[30].

El modelo se entrenó y validó en una granja ubicada en Ozzano Emilia Romagna, Bolonia (Italia), empleando una red neuronal YOLO, específicamente diseñada para el reconocimiento automático de bovinos en videos obtenidos dentro del establo. Para el entrenamiento y validación se seleccionaron cuatro vacas de la raza Holstein Friesian, utilizando el patrón característico de su pelaje como criterio distintivo[30].



Figura 12. Visión artificial aplicada en bovinos Holstein Friesian [30]

Los resultados mostraron una precisión promedio entre 0,64 y 0,66, lo que evidencia la viabilidad de identificar individuos a partir de su morfología y de la disposición particular de las manchas del pelaje, las cuales representan un rasgo claramente diferenciable para los sistemas de visión artificial.

En el año 2022 se publicó un estudio en el que se propusieron estrategias de procesamiento de imágenes orientadas a la estimación del peso vivo (PV) de cerdos, considerado un parámetro fundamental para el seguimiento de la ganancia diaria, el estado nutricional, el desempeño sanitario y la proyección del peso comercial.

El enfoque se basa en un método sin contacto que emplea las dimensiones físicas del animal, estableciendo correlaciones con su PV con el fin de reducir el estrés asociado a los procedimientos tradicionales. Tras la adquisición de las imágenes, se aplican diversas etapas de procesamiento, entre ellas filtrado, extracción de características, así como el entrenamiento y la construcción de bases de datos, lo que permite obtener una predicción confiable del peso vivo[30].

Tras la adquisición de las imágenes, se aplican diversas etapas de procesamiento que garantizan la calidad y utilidad de la información capturada. Entre estas fases se incluyen el preprocesamiento y filtrado de las imágenes para eliminar ruidos o distorsiones, la segmentación para aislar al animal del fondo, y la extracción de características morfométricas relevantes, tales como el área proyectada, la longitud corporal, la circunferencia torácica y la altura a la cruz. Posteriormente, estos parámetros se utilizan como variables de entrada en algoritmos de regresión o modelos de aprendizaje automático que permiten establecer correlaciones robustas con el peso real del animal[31].

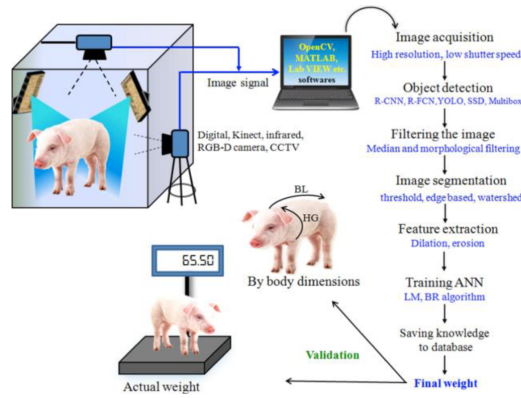


Figura 13. Procesamiento para entrenamiento y formación de base de datos [30]

Finalmente, el entrenamiento de dichos modelos, acompañado de la construcción de bases de datos representativas, posibilita la obtención de predicciones confiables del peso vivo, con márgenes de error reducidos frente a métodos convencionales. Esto no solo contribuye a una gestión más eficiente de la producción porcina, sino que también sienta las bases para el desarrollo de sistemas automatizados de monitoreo en granjas, alineados con los principios del bienestar animal y de la producción pecuaria sostenible[31].

Este tipo de metodologías no se limitan únicamente a la especie porcina, sino que han sido adaptadas y exploradas en otras especies de interés zootécnico. Por ejemplo, en bovinos de carne y leche, se han empleado cámaras RGB y sensores de profundidad (como Microsoft Kinect o Intel RealSense) para estimar el peso corporal a través de mediciones de ancho, altura y área de superficie dorsal. Estos métodos han demostrado alta precisión, reduciendo significativamente la necesidad de movilizar a los animales hacia corrales de pesaje, lo cual disminuye el riesgo de lesiones tanto para el ganado como para los operarios[32].

En el caso de las aves de corral, el enfoque ha sido similar, aunque con adaptaciones específicas debido al tamaño y la movilidad de los animales. Investigaciones recientes han utilizado imágenes aéreas tomadas desde cámaras instaladas en el techo de los galpones para estimar el peso de pollos de engorde. Los modelos de visión por computadora analizan la silueta y la densidad de los animales dentro del grupo, generando predicciones individuales y grupales del PV, lo que facilita la toma de decisiones en tiempo real sobre alimentación y manejo productivo[32].

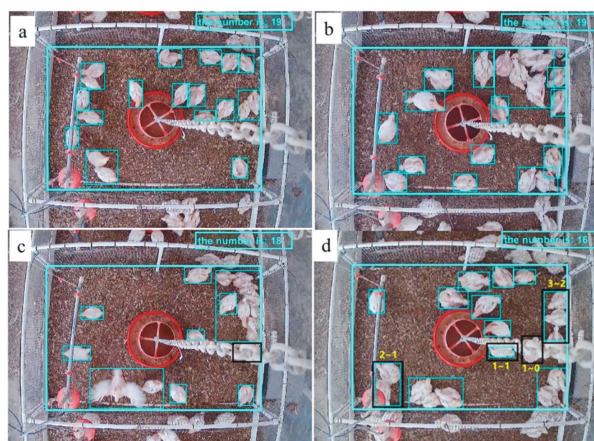


Figura 14. Estimación de numero de aves mediante procesamiento de imágenes [31]

V-C. Algoritmos y modelos de detección de objetos en visión artificial

La detección de objetos es uno de los campos de mayor crecimiento dentro de la visión por computadora y el aprendizaje profundo. Su relevancia radica en la combinación de dos tareas fundamentales: la clasificación de objetos, que permite identificar a qué categoría pertenece cada elemento, y la localización de objetos, que determina en qué parte de la imagen se encuentran dichos elementos. Estas capacidades hacen que la detección sea un área clave para el desarrollo de sistemas de visión artificial aplicados a la agricultura, ganadería, seguridad, automatización industrial, entre otros[33].

En el contexto de esta investigación, resulta importante analizar los algoritmos de detección más representativos, los cuales han marcado un antes y un después en el desarrollo de aplicaciones prácticas de visión artificial:

- Faster R-CNN

El modelo Faster R-CNN, desarrollado por investigadores de Microsoft, constituye una de las arquitecturas más influyentes en la evolución de la detección de objetos. Está compuesto por dos módulos principales: la Red de Propuestas de Regiones (RPN), que se encarga de sugerir posibles zonas donde pueden encontrarse objetos, y la red Fast R-CNN, que clasifica dichas regiones y genera sus correspondientes cajas delimitadoras (bounding boxes). Este enfoque de dos etapas ofrece una alta precisión en la detección, aunque con un costo computacional mayor que otros algoritmos más recientes. No obstante, Faster R-CNN sirvió de base para el desarrollo de modelos posteriores y aún es utilizado en escenarios donde la precisión es prioritaria sobre la velocidad[33].

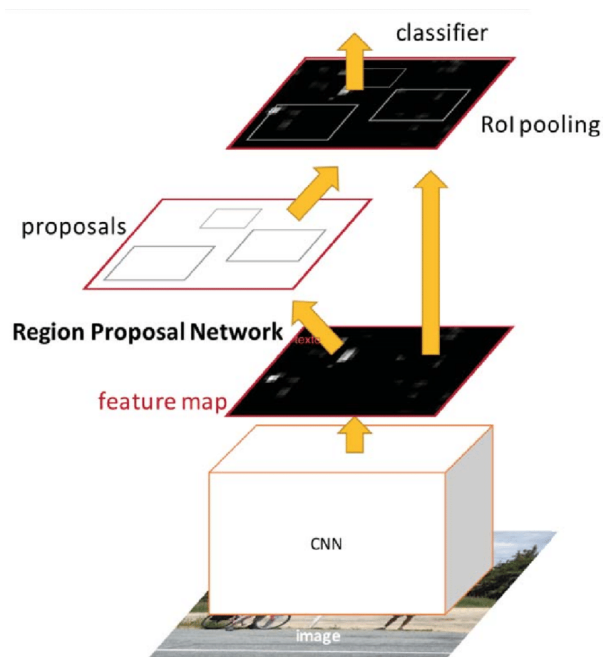


Figura 15. Modelo desarrollado por microsoft Faster R-CNN [33]

- YOLO (You Only Look Once)

El algoritmo YOLO supuso un cambio significativo al introducir un enfoque de detección en tiempo real. A diferencia de los modelos de dos etapas, YOLO procesa la imagen completa en una sola pasada, dividiéndola en una cuadrícula y asignando a cada celda la responsabilidad de predecir cajas delimitadoras y clases. Su principal ventaja es la velocidad, alcanzando tasas de detección cercanas al tiempo real, sin sacrificar una

precisión considerable. Gracias a esto, YOLO se ha consolidado como uno de los modelos más utilizados en aplicaciones prácticas, como la detección de animales en el campo, el monitoreo de cultivos y la vigilancia mediante cámaras inteligentes[33].

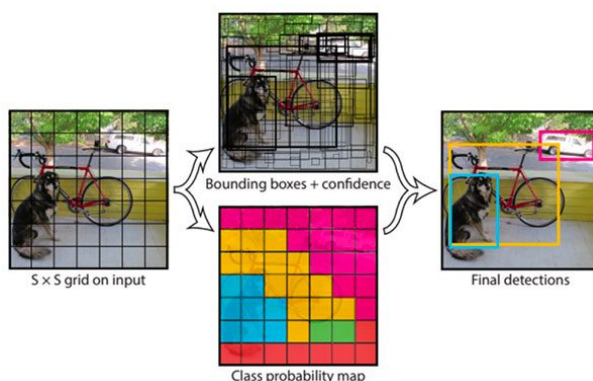


Figura 16. Modelo YOLO enfocado en la detección en tiempo real [33]

- SSD (Single Shot Detector)

El modelo SSD también pertenece a la categoría de detectores de una sola pasada, lo que le otorga rapidez en la predicción de objetos. A diferencia de YOLO, SSD se basa en la generación de predicciones desde múltiples mapas de características a diferentes escalas, lo que le permite reconocer objetos de diversos tamaños con buena precisión. Al no requerir una red de propuestas de regiones, SSD ofrece un equilibrio entre velocidad y precisión, resultando adecuado para aplicaciones que demandan procesamiento en tiempo real sin un consumo excesivo de recursos[33].

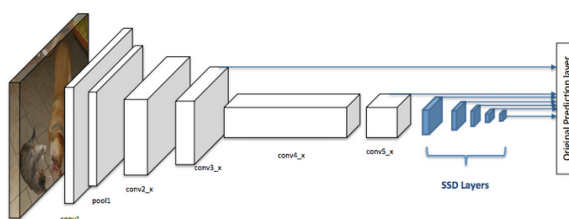


Figura 17. Estructura general del modelo SSD (Single Shot Detector) utilizado para la detección de objetos en una sola pasada. [33]

V-C1. Comparación y aplicaciones: Al analizar estos tres algoritmos, puede observarse que Faster R-CNN destaca por su precisión, mientras que YOLO y SSD sobresalen en escenarios donde la rapidez es fundamental. En particular, YOLO y SSD han demostrado ser más adecuados para proyectos en los que se requiere el procesamiento inmediato de datos visuales, como la clasificación de estados de maduración en productos agrícolas o el monitoreo de animales en movimiento.

La evolución de estos modelos no solo ha fortalecido la detección de objetos en 2D, sino que también ha abierto la puerta a nuevas investigaciones en segmentación semántica, detección 3D y sistemas multimodales, que siguen construyendo sobre los avances iniciales de estas arquitecturas.

V-D. Redes Neuronales Convolucionales (CNNs) para el Análisis de Imágenes

Las Redes Neuronales Convolucionales (CNN) constituyen una arquitectura de aprendizaje profundo creada específicamente para manejar el procesamiento de datos de cuadrícula, como las imágenes. Su arquitectura se

inspira en la organización de la corteza visual animal y se compone de capas fundamentales:

Capas Convolucionales: Estas capas aplican filtros (pequeñas matrices de pesos) a la imagen de entrada, realizando operaciones de convolución para extraer características espaciales jerárquicas, como bordes, esquinas, texturas y patrones más complejos a medida que la profundidad de la red aumenta [5].

Tienen eficacia en tareas de visión por computadora, como:

- Clasificación de imágenes
- Detección de objetos
- Segmentación de imágenes
- Reconocimiento facial
- Diagnóstico médico por imágenes

V-D1. Estructura básica de una CNN: Una red convolucional típica se compone de los siguientes bloques principales:

1. Capa de convolución (Convolutional Layer)

Aplica un filtro (o kernel) sobre la imagen para extraer características locales, como bordes, texturas o esquinas.

- Usa operaciones de convolución, no multiplicación de matrices como en las redes tradicionales.
- El resultado es un mapa de activación o feature map.

2. Función de activación (ReLU)

Se suele aplicar ReLU (Rectified Linear Unit) después de la convolución para introducir no linealidad, tal como en la siguiente ecuación:

$$f(x) = \text{máx}(0, x) \quad (1)$$

3. Capas de Pooling (Agrupación)

Estas capas reducen la dimensionalidad espacial de las representaciones aprendidas en las capas convolucionales, lo que ayuda a disminuir la cantidad de parámetros, controlar el sobreajuste y hacer que las características sean más robustas a pequeñas variaciones en la posición y la escala de los objetos[7].

- Ejemplo común: Max pooling (elige el valor máximo de una región).
- Ayuda a reducir el número de parámetros y el riesgo de sobreajuste.

4. Capas completamente conectadas (Fully Connected Layers)

Después de extraer características, se aplanan el resultado y se pasa por una o más capas densamente conectadas para hacer la clasificación final. Al final de la red, las características extraídas por las capas convolucionales y de pooling se aplanan y se conectan a capas totalmente conectadas, que realizan la tarea de clasificación final, asignando probabilidades a diferentes clases (en tu caso, los diferentes estados de maduración del cacao).

5. Capa de salida

Suele ser una Softmax para clasificación multiclase, generando una probabilidad por clase.

Funciones de Activación: Las funciones de activación se utilizan tras cada capa convolucional y con frecuencia después de las capas totalmente conectadas con el fin de incorporar no linealidad a la red, lo que le permite captar relaciones complejas presentes en los datos. Entre las más empleadas se encuentran ReLU (Rectified Linear Unit) y sus diferentes variantes. [34].

La capacidad de las CNNs para aprender automáticamente representaciones jerárquicas de las características visuales las ha convertido en la base de muchos sistemas exitosos de clasificación y detección de objetos en

imágenes como se muestra en la figura 18.

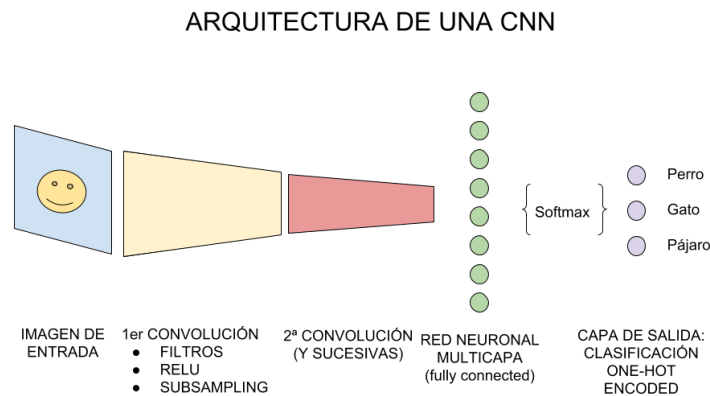


Figura 18. Estructura de una CNN [35]

V-D2. Aplicación de CNN en la Clasificación de Maduración del Cacao: La utilización de CNN para clasificar el estado de maduración del cacao implica entrenar el modelo con un conjunto representativo de imágenes etiquetadas. El modelo aprende a identificar los patrones visuales (color, textura, forma) asociados con cada nivel de maduración.

1. Fases del desarrollo del sistema

- Recolección de imágenes: captura de imágenes de frutos de cacao en diferentes etapas de maduración.
- Etiquetado de datos: asignación de clases (verde, maduro, pasado, etc.) a cada imagen.
- Preprocesamiento: redimensionamiento, normalización y aumento de datos (data augmentation).
- Entrenamiento del modelo: se utiliza una arquitectura CNN, que puede ser personalizada o basada en modelos preentrenados (como MobileNet, VGG16 o ResNet).
- Evaluación del modelo: validación de su desempeño utilizando métricas como precisión, exactitud, sensibilidad y matriz de confusión.
- Implementación práctica: se puede integrar en una aplicación web, móvil o embebida en sistemas físicos (como una Raspberry Pi con cámara).

V-D3. Retos y Consideraciones: Aunque las CNN son altamente efectivas, la implementación de un sistema de visión artificial en campo debe considerar algunos desafíos:

- Variaciones en la iluminación: luz solar directa, sombra o condiciones climáticas pueden alterar la imagen.
- Diversidad morfológica del fruto: diferentes variedades de cacao presentan variaciones en su forma y color.
- Ruido de fondo: hojas, ramas, u otros frutos pueden interferir en la predicción.
- Balance de datos: es importante tener una cantidad equilibrada de imágenes por clase para evitar sesgos en el modelo.

V-E. Google Colab

Google Colaboratory (Colab) es una plataforma gratuita basada en cuadernos Jupyter que se ejecuta en la nube y ofrece acceso sin costo a unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPUs). Esto lo convierte en una plataforma invaluable para investigadores y desarrolladores de aprendizaje automático, ya

que acelera significativamente el entrenamiento de modelos computacionalmente intensivos, como las CNNs. Colab facilita la carga y manipulación de conjuntos de datos, la escritura y ejecución de código Python, y la visualización de resultados [36] .

Una herramienta esencial para entrenamiento de modelos sin necesidad de tener una GPU local. Permite usar notebooks de Python en la nube con acceso gratuito a recursos acelerados como GPU y TPU. Es ideal para estudiantes y proyectos de tesis.



Figura 19. Google Colab [36]

V-F. Labelling

El labelling o anotación de datos es una etapa importante en el desarrollo de modelos de aprendizaje supervisado, especialmente para tareas de visión artificial como la detección de objetos. Para la detección del estado de maduración del cacao, esto implica típicamente dibujar cajas delimitadoras alrededor de cada fruto de cacao presente en la imagen y asignarle una etiqueta que corresponda a su grado de maduración (“maduro”, “inmaduro”, “sobremaduro”).

La calidad del conjunto de datos etiquetado tiene un impacto directo en el rendimiento del modelo entrenado. Un labelling preciso, consistente y exhaustivo es fundamental para que el modelo aprenda a reconocer correctamente las características visuales distintivas de cada estado de maduración . Se deben considerar aspectos como la precisión en el trazado de las cajas delimitadoras, la correcta asignación de etiquetas y la cobertura de la variabilidad de los frutos en diferentes condiciones de iluminación, ángulos de visión y etapas de maduración.

Existen diversas herramientas de labelling disponibles, tanto de escritorio como en línea, que facilitan este proceso. La elección de la herramienta adecuada dependerá de factores como el tamaño del conjunto de datos, la complejidad de las anotaciones requeridas y los recursos disponibles [37]

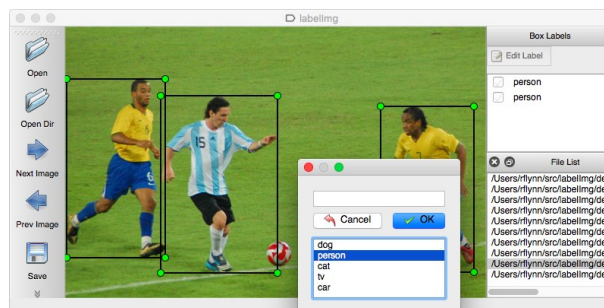


Figura 20. labelling [37]

V-G. Visual Studio 2019

Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que ofrece una plataforma completa para que los programadores puedan escribir, depurar, compilar y ejecutar aplicaciones de forma centralizada. Se podría decir que es como una “caja de herramientas digital” para desarrolladores de software.

Visual Studio está diseñado principalmente para crear aplicaciones de escritorio, móviles, web y servicios en la nube. Es compatible con muchos lenguajes de programación, como: Python, C++, C#, JavaScript, HTML/CSS, .NET (framework de Microsoft). Esto hace que sea extremadamente versátil, tanto para principiantes como para desarrolladores avanzados [38].

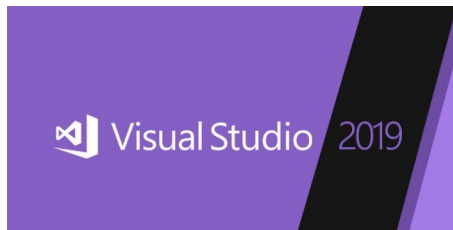


Figura 21. Visual Studio 2019 [38]

V-H. Redes Pre-entrenadas y Transfer Learning

Las redes neuronales pre-entrenadas son modelos que han sido entrenados en conjuntos de datos masivos y genéricos, como ImageNet, que contiene millones de imágenes de miles de clases diferentes. Estos modelos han aprendido a extraer características visuales generales que son útiles para una amplia gama de tareas de reconocimiento de imágenes, como la detección de bordes, texturas, formas y objetos básicos.[39]

El Transfer Learning es una técnica de aprendizaje automático que consiste en utilizar el conocimiento adquirido por un modelo pre-entrenado como punto de partida para entrenar un nuevo modelo en una tarea diferente pero relacionada con un conjunto de datos más pequeño. En el contexto de la detección del estado de maduración del cacao, se podría utilizar un modelo pre-entrenado en ImageNet y ajustar sus capas superiores para que aprendan a clasificar los diferentes estados de maduración del cacao utilizando un conjunto de datos específico de frutos de cacao etiquetados.

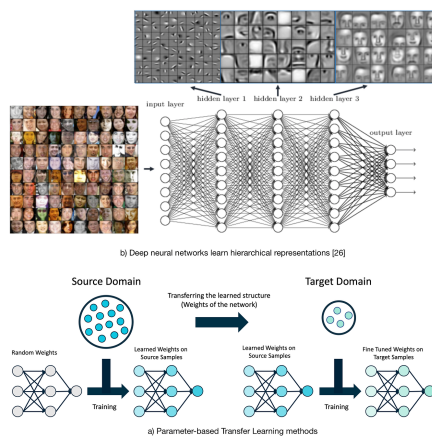


Figura 22. Arquitectura del Transfer Learning [40]

VI. MARCO METODOLÓGICO

VI-A. Adquisición de la imagen

Consiste en capturar imágenes de las mazorcas de cacao utilizando dispositivos como cámaras digitales. Estas imágenes servirán como base para el análisis posterior y se reunirá una colección de fotografías de cacao en distintas fases de maduración tanto inmaduro, maduro y sobre maduro respectivamente. Es importante tomar a consideración la iluminación y que la resolución de la imagen sea lo suficientemente buena, para que características visuales y relevantes como el color y la textura puedan ser reconocidas con mayor facilidad, una clave para asegurar variedad en las imágenes y que sea más fácil el entrenamiento del modelo es variar las formas de iluminación, los ángulos de toma y los fondos para que el sistema sea robusto en diversas condiciones. Cada imagen será registrada con su origen y, lo más importante, su clasificación, estas mismas que son asignadas manualmente dependiendo de su estado de maduración.

VI-B. Clases del dataset elegido

Tabla II
CANTIDAD DE IMÁGENES POR CLASE DE MADURACIÓN DEL CACAO. AUTORÍA PROPIA.

Clase	Descripción	Nº de Imágenes
Inmaduro	Fruto rojizo o verde normalmente de tamaño pequeño.	1000
Maduro	Fruto amarillo y sin pintas de otro color.	900
Sobre maduro	Cáscara deteriorada y con indicios marrones.	800
Total		2700

VI-C. Preprocesamiento

En esta fase, se aplican técnicas para mejorar la calidad de las imágenes y facilitar su análisis. Las operaciones incluyen:

- Corrección de iluminación: Ajuste del brillo y contraste para compensar variaciones en la iluminación durante la captura.
- Reducción de ruido: Aplicación de filtros, como el filtro gaussiano, para eliminar imperfecciones o artefactos no deseados.
- Normalización de color: Estandarización de los colores para asegurar consistencia entre diferentes imágenes.
- Recorte y redimensionamiento: Ajuste del tamaño y enfoque de las imágenes para centrar la atención en las mazorcas.

VI-D. Segmentación

La segmentación implica dividir la imagen en regiones significativas, separando las mazorcas del fondo y de otros elementos. Esto se puede lograr mediante técnicas como la umbralización por color, que identifica áreas con colores específicos asociados a las mazorcas, o mediante algoritmos de detección de bordes, que resaltan los contornos de las mazorcas. La segmentación precisa es importante para enfocar el análisis en las áreas relevantes de la imagen.

VI-E. Extracción de características

Una vez segmentadas las mazorcas, se extraen características visuales que son indicativas de su estado de maduración. Estas características pueden incluir:

- Color: Análisis de los valores de color en diferentes espacios de color (RGB, HSV) para identificar cambios asociados a la madurez.
- Textura: Evaluación de patrones de textura en la superficie de la mazorca que pueden variar con la maduración.
- Forma y tamaño: Medición de la geometría de las mazorcas, ya que ciertos cambios morfológicos pueden estar relacionados con la madurez.

VI-F. Etiquetación

Durante el entrenamiento en Google Colab, se empleó la función `flow_from_directory()` de `ImageDataGenerator`, la cual recorre automáticamente la estructura de carpetas del conjunto de datos y asigna un índice numérico a cada clase según el orden alfabético en que se encuentran. Dichos índices son transformados internamente en vectores *one-hot encoding*, formato que permite que las imágenes sean interpretadas correctamente por la red neuronal.

Esto eliminó la necesidad de generar manualmente un archivo de etiquetas externo, lo que facilita en la asignación de clases y la organización de los datos.

VI-G. Clasificación

Con las características extraídas, se emplean algoritmos de aprendizaje automático para clasificar las mazorcas en diferentes categorías de madurez como, fructificación, desarrollo, madurez. Modelos como redes neuronales convolucionales (CNN) o algoritmos de clasificación supervisada pueden ser entrenados con conjuntos de datos etiquetados para realizar esta tarea con alta precisión.

Tabla III
ARQUITECTURA DE LA RED NEURONAL CONVOLUCIONAL PROPUESTA

Capa	Tipo	Filtros/Unidades	Tamaño Kernel	Activación	Salida
Entrada	Imagen	-	-	-	128x128x3
Capa 1	Conv2D	32	3x3	ReLU	126x126x32
Capa 2	MaxPooling2D	-	2x2	-	63x63x32
Capa 3	Conv2D	64	3x3	ReLU	61x61x64
Capa 4	MaxPooling2D	-	2x2	-	30x30x64
Capa 5	Flatten	-	-	-	57600
Capa 6	Dense	128	-	ReLU	128
Capa 7	Dense	4	-	Softmax	4

VI-G1. Parámetros del modelo CNN:

VI-H. Captura de imágenes y visualización de resultados

Para la fase de interacción con la aplicación, la captura de imágenes se realizará cargando fotografías previamente ya tomadas desde un dispositivo móvil. Los usuarios podrán seleccionar una imagen desde la galería, a través de la interfaz web que abrirán en Google Chrome. Una vez enviada la foto, el sistema de visión artificial la analizará para determinar el estado de maduración del cacao. De acuerdo con ello, la interfaz ofrecerá un análisis de la imagen y una confiabilidad porcentual en la que nos dirá que tan confiable es el resultado que nos entrega el sistema de visión artificial.

La visualización de resultados consiste en superponer recuadros alrededor de cada mazorca detectada en la imagen y mostrar la etiqueta de clase junto con la probabilidad de la predicción. Esto se puede implementar dibujando

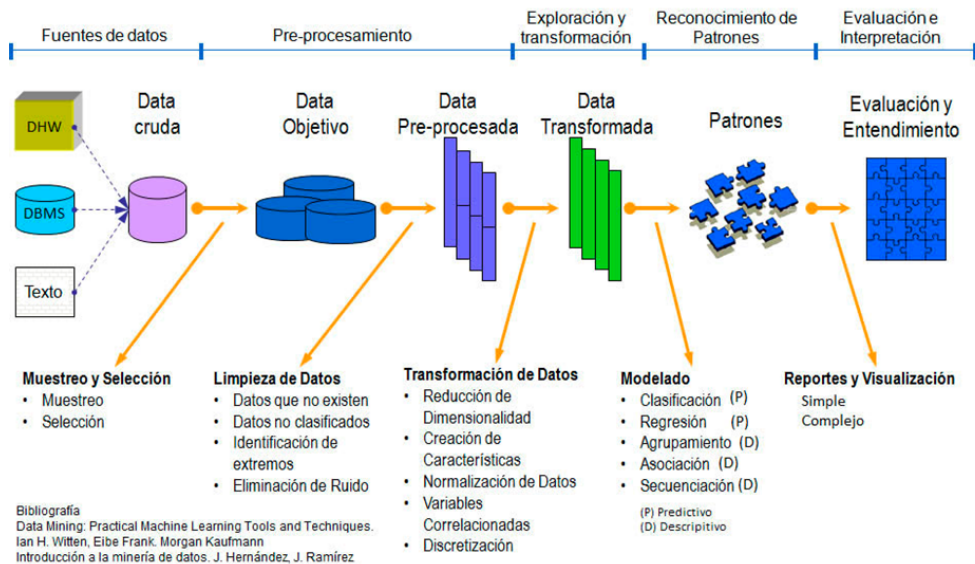


Figura 23. Procesamiento de imágenes con python. Autoría propia

sobre un Canvas o utilizando vistas de imagen con gráficos. Adicionalmente, la aplicación mostrará un resumen de resultados en un panel de texto. El flujo completo es interactivo: tras la captura, el usuario ve inmediatamente en pantalla las cajas y etiquetas generadas, facilitando la interpretación en campo.

VI-I. Importación de librerías y montaje del entorno

En esta parte se empezó por hacer el código en Google colab inicial del código se están importando las bibliotecas necesarias para ejecutar todo el proyecto, **numpy** (abreviado como np) permite realizar operaciones matemáticas sobre matrices, que son la base para procesar imágenes digitales. Si se tiene una imagen de una mazorca, internamente no es más que una matriz de números que representan colores y tonalidades.

Luego, **pandas (pd)** permite trabajar con datos en formato de tabla, aunque en este script no se usa directamente **.os** y **glob** son herramientas para moverse entre carpetas y buscar archivos automáticamente; si se necesitase cargar todas las imágenes dentro de **/train/Maduro**, puedes usar glob para que las recoja automáticamente, sin tener que escribir cada ruta a mano.

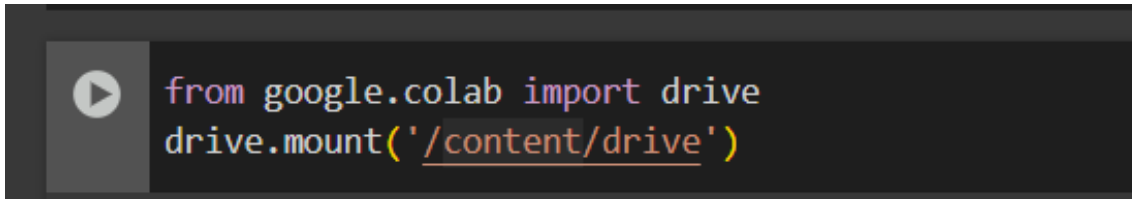
También se importa **matplotlib.pyplot (plt)**, una librería para hacer gráficos, por ejemplo, una curva de precisión del modelo durante el entrenamiento. Finalmente, se monta Google Drive con **drive.mount('/content/drive')**, lo que permite acceder a los archivos personales como si fueran parte del sistema local de archivos en Google Colab, como se puede apreciar en la figura 24.

```
[ ] # General imports
import numpy as np
import pandas as pd
import os
from glob import glob
import matplotlib.pyplot as plt
```

Figura 24. Importación de librerías. Autoría propia

VI-J. Definición de rutas y conteo de imágenes

En este bloque definimos dos rutas: **train_path** y **valid_path**, que apuntan a las carpetas de entrenamiento y validación dentro del Google Drive. Cada una de estas carpetas contiene, a su vez, tres subcarpetas llamadas Inmaduro, Maduro y Sobre maduro; las imágenes se almacenan allí para que Keras pueda reconocer automáticamente qué clase pertenece a cada archivo, como en la figura 25.



```
from google.colab import drive
drive.mount('/content/drive')
```

Figura 25. Definición de rutas y conteo de imágenes. Autoría propia

Con **os.listdir()** obtenemos la lista de archivos que cuelgan directamente de cada carpeta y medimos su longitud con **len()** como se observa en la figura 26.

```
[ ] train_path = '/content/drive/MyDrive/train'
valid_path = '/content/drive/MyDrive/valid'
```

Figura 26. Definición de rutas. Autoría propia

En la figura 27 como se observa que esto nos da un conteo rápido para comprobar que efectivamente existen imágenes y que la proporción entre entrenamiento y validación es razonable, por ejemplo $\approx 70\%$ para entrenar y 30% para validar.

```
[ ] train_images = glob(os.path.join(train_path, '/content/drive/MyDrive/train/Inmaduro')
valid_images = glob(os.path.join(valid_path, '/content/drive/MyDrive/valid/Inmaduro')
#print opcional para saber que si me reconoce las carpetas
print("Número de imágenes en la carpeta de entrenamiento:", len(train_images))
print("Número de imágenes en la carpeta de prueba:", len(valid_images))
```

Figura 27. conteo de imágenes. Autoría propia

VI-K. *Recolección de rutas de imágenes*

Este bloque intenta cargar directamente las rutas de todas las imágenes dentro de las carpetas de entrenamiento y validación, para asegurarse de que las rutas están correctamente escritas y que no hay errores de lectura. **glob** permite buscar archivos usando expresiones con comodines, útil cuando se tiene una estructura de carpetas y subcarpetas como se observa en la figura 28.

```
# Leer el número de imágenes en las carpetas de entrenamiento y prueba
num_train_images = len(os.listdir(train_path))
num_test_images = len(os.listdir(valid_path))
print("Número de imágenes en la carpeta de entrenamiento:", num_train_images)
print("Número de imágenes en la carpeta de prueba:", num_test_images)
```

Figura 28. Recolección de rutas de imágenes. Autoría propia

Aunque en este caso específico no se utiliza directamente esta lista de rutas, es útil para depuración y verificación previa.

VI-L. Generadores de datos para entrenamiento y validación

Como se observa en la figura 29 se identifica los siguientes:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
```

Figura 29. Entrenamiento y validación del modelo. Autoría propia

- ImageDataGenerator: Genera lotes de imágenes en tiempo real desde carpetas. También permite aplicar técnicas de aumento de datos (data augmentation) como rotación, zoom, etc.
- Sequential: Permite construir una red neuronal capas por capas de manera lineal. Es ideal para arquitecturas simples como esta.
- Adam: Es un algoritmo de optimización que ajusta automáticamente la tasa de aprendizaje para acelerar y estabilizar el aprendizaje.

Si las imágenes no están bien normalizadas, alineadas en tamaño o correctamente etiquetadas, el modelo nunca aprenderá correctamente o tendrá un rendimiento pobre. Toda la inteligencia del sistema que se está desarrollando en este caso, para clasificar el grado de madurez del cacao depende completamente de qué tan bien se realice esta etapa de preprocesamiento. Además, al usar herramientas como ImageDataGenerator, se automatiza gran parte del trabajo manual, se mejora la eficiencia del código y se facilita la implementación del modelo en otras plataformas.

Este tipo de preparación se realiza tanto para los datos de entrenamiento como para los de validación. En el entrenamiento, los datos se mezclan aleatoriamente en cada época para evitar que el modelo memorice el orden de las imágenes, y así generalice mejor según la imagen 30 .

```
# Generador para entrenamiento
train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical', # ← Cambiado de 'binary' a 'categorical'
    shuffle=True # Mezcla aleatoria para mejor entrenamiento
)
```

Figura 30. Entrenamiento y validación del modelo

En la validación, no se mezclan, porque se necesita un conjunto fijo para comparar y medir objetivamente qué tan bien está aprendiendo el modelo como se logra ver en la figura 31.

```
# Generador para validación
valid_datagen = ImageDataGenerator(rescale=1./255)

valid_generator = valid_datagen.flow_from_directory(
    valid_path,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False # No es necesario mezclar en validación
)

# Mostrar clases detectadas
print("Clases detectadas automáticamente:")
print(train_generator.class_indices)
```

Figura 31. Entrenamiento y validación del modelo. Autoría propia

VI-M. Definición y compilación del modelo de red neuronal convolucional (CNN)

En esta etapa se lleva a cabo la definición de la arquitectura del modelo de aprendizaje profundo, que será la parte importante del sistema de clasificación automática de mazorcas de cacao según su grado de maduración: inmaduro, maduro o sobremaduro.

El modelo fue definido como una estructura secuencial (Sequential), lo que implica que las capas se apilan una tras otra, en un orden lineal. Esta elección es adecuada para tareas de clasificación en las que se parte de una imagen de entrada y se desea obtener una etiqueta única como salida así como se observa en la figura 32.

```
# Definir el modelo
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),

    Flatten(),

    Dense(128, activation='relu'),
    Dropout(0.5),

    Dense(3, activation='softmax') # 3 clases de madurez del cacao
])
```

Figura 32. compilación del modelo CNN. Autoría propia

La arquitectura del modelo consta de varias capas esenciales:

- **Capas convolucionales (Conv2D):** Estas capas tienen la función de detectar características espaciales relevantes dentro de las imágenes, como bordes, texturas o patrones específicos en la mazorca. Se emplean tres bloques convolucionales, aumentando progresivamente la cantidad de filtros (32, 64 y 128), lo que permite al modelo aprender representaciones de mayor complejidad en cada etapa.
- **Capas de reducción de dimensionalidad (MaxPooling2D):** Después de cada capa de convolución se emplea una capa adicional de 'max pooling'. Esta técnica permite disminuir la dimensión espacial sacadas de las características, lo que disminuye la carga computacional y ayuda a evitar el sobreajuste, al tiempo que conserva la información más importante.
- **Capa de aplanamiento (Flatten):** Esta capa transforma las matrices de características extraídas en un vector unidimensional, lo que es necesario para poder conectar con las capas densas (fully connected) finales que realizarán la clasificación.
- **Capa densa oculta (Dense):** Se añade una capa totalmente conectada compuesta por 128 neuronas, utilizando como función de activación ReLU (Rectified Linear Unit). Esta capa permite realizar combinaciones no lineales de las características detectadas por las capas anteriores, ayudando a afinar la decisión de clasificación.
- **Capa de regularización (Dropout):** Se incorpora una capa Dropout con una tasa del 50 %, cuyo objetivo es prevenir el sobreajuste (overfitting) mediante la eliminación aleatoria de la mitad de las conexiones durante el proceso de entrenamiento.
- **Capa de salida:** Finalmente, la capa de salida cuenta con 3 neuronas (una por cada clase: inmaduro, maduro, sobremaduro), y utiliza una función de activación softmax, que convierte las salidas en una distribución de probabilidad para cada clase. El modelo seleccionará la clase con mayor probabilidad como su predicción final tal y como se observa en la figura 33.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_4 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_5 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_4 (Dense)	(None, 128)	4,735,104
dropout_1 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 3)	387

Total params: 4,828,739 (18.42 MB)
 Trainable params: 4,828,739 (18.42 MB)
 Non-trainable params: 0 (0.00 B)

Figura 33. compilación del modelo CNN. Autoría propia

Tras definir la arquitectura, el modelo se compila utilizando el optimizador Adam con una tasa de aprendizaje de 0.0001. Para la función de pérdida se emplea `categorical_crossentropy`, adecuada para problemas de clasificación multiclase con etiquetas codificadas en formato one-hot. Como métrica de evaluación se consideró la precisión, por ser un indicador claro del porcentaje de aciertos del modelo durante el entrenamiento y validación mostrados en la figura 34.

```
# Compilar el modelo
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy', # Porque usamos etiquetas one-hot
    metrics=['accuracy']
)

# Mostrar resumen del modelo
model.summary()
```

Figura 34. compilación del modelo CNN. Autoría propia

VI-N. Entrenamiento del modelo

Ya definida y compilada la arquitectura de la red neuronal convolucional (CNN) como se observa en la figura 35, se procede a su entrenamiento, que representa una de las etapas más importantes del desarrollo del sistema inteligente de clasificación del grado de maduración del cacao.

```
# Entrenar el modelo
history = model.fit(
    train_generator,
    epochs=10, # Puedes ajustar según tu tiempo/datos
    validation_data=valid_generator
)
```

Figura 35. Entrenamiento del modelo. Autoría propia

- **train_generator:** Este parámetro representa el conjunto de datos de entrenamiento, el cual ha sido previamente preparado con `ImageDataGenerator`. Este generador alimenta al modelo con lotes de imágenes preprocesadas, en este caso, normalizadas (escaladas entre 0 y 1) y con sus respectivas etiquetas categóricas codificadas.
- **epochs=10:** Esta instrucción indica el número de veces que el modelo verá todo el conjunto de entrenamiento completo. Cada pasada completa se denomina época. El valor de 10 sirve como punto de partida para tareas con conjuntos de datos de tamaño medio, y permite observar el comportamiento del modelo sin un entrenamiento excesivo que podría llevar al sobreajuste (`overfitting`).
- **validation_data=valid_generator:** El uso de un conjunto de validación independiente durante el entrenamiento es fundamental para evaluar el rendimiento del modelo en datos que no ha visto previamente.

Esto permite monitorear si el modelo está aprendiendo patrones útiles o simplemente memorizando las imágenes del entrenamiento. El generador de validación está configurado de manera similar al de entrenamiento, pero sin mezcla de datos (`shuffle=False`), para mantener un orden y reproducibilidad en los resultados.

A mayor número de épocas, más tiempo de entrenamiento, pero también mayor riesgo de sobreentrenamiento (memorizar en lugar de generalizar).

Durante el entrenamiento, en cada época, el modelo realiza dos tareas:

- **Forward pass:** Pasa las imágenes por la red neuronal y genera una predicción.
- **Backpropagation:** Compara la predicción con la etiqueta real (usando la función de pérdida) y ajusta internamente los pesos del modelo con el optimizador Adam.

Cada iteración dentro de una época procesa un lote de imágenes (de tamaño definido por `batch_size`), y al final de cada época se reportan dos métricas como se observa en la figura 36:

- Precisión (`accuracy`) y pérdida (`loss`) del entrenamiento, que indican qué tan bien el modelo se ajusta a los datos que ha visto.
- Precisión y pérdida de validación, que muestran el desempeño sobre datos nuevos.

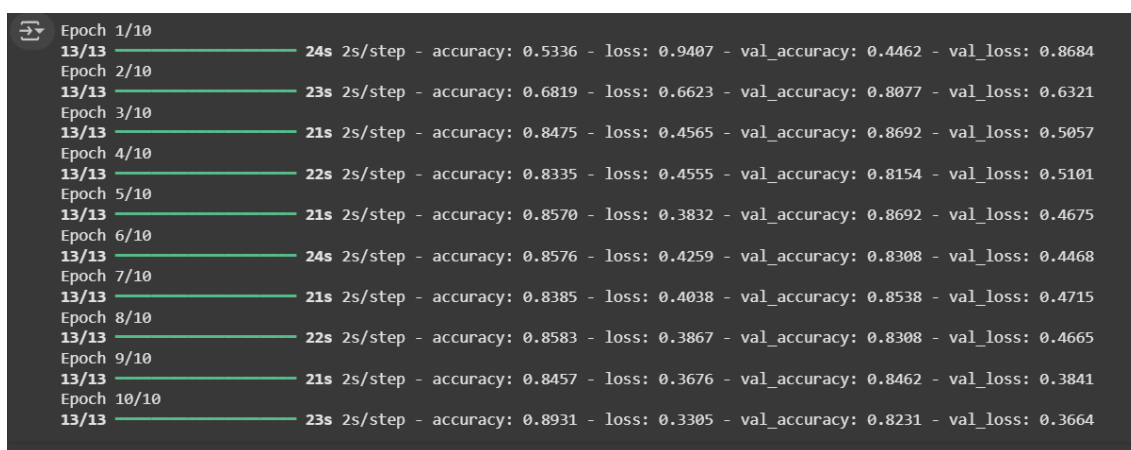


Figura 36. Entrenamiento del modelo en 10 épocas. Autoría propia

Esta imagen muestra el registro del entrenamiento del modelo de red neuronal convolucional (CNN) durante 10 épocas. Específicamente, corresponde al proceso de ajuste de pesos internos de la red mientras aprende a clasificar imágenes de mazorcas de cacao en tres clases: Inmaduro, Maduro y Sobremaduro. Cada línea indica una época del entrenamiento. En total hay 10, lo cual significa que el modelo ha revisado todo el conjunto de datos 10 veces, a continuación se explica que realiza el modelo en cada época:

▪ Época 1

accuracy: 0.5336 → El modelo acierta el 53 % en entrenamiento.

val_accuracy: 0.4462 → Solo acierta el 44 % en validación.

El modelo aún está aprendiendo. La pérdida es alta (cercana a 1), indicando predicciones imprecisas.

▪ Época 2

Mejora significativa: **accuracy** sube a 68 % y **val_accuracy** a 80 %.

Esto muestra que el modelo empieza a capturar patrones relevantes en las imágenes.

- **Épocas 3 a 5**

El modelo alcanza valores de hasta 85 % de precisión en entrenamiento y 86–87 % en validación.

El modelo no solo memoriza, sino que generaliza bien a nuevos datos.

Las pérdidas (**loss**) también disminuyen considerablemente, lo cual confirma este comportamiento.

- **Épocas 6 a 10**

Precisión se mantiene estable o mejora levemente: hasta **accuracy**: 0.8931 y **val_accuracy**: 0.8231.

Las pérdidas también se reducen, llegando hasta **val_loss**: 0.3664, un valor bastante bajo.

VI-Ñ. Visualización del Rendimiento del Modelo

Una vez completado el proceso de entrenamiento del modelo de red neuronal convolucional, se evalúa el rendimiento. Para esto, se utilizan gráficos que permitan observar cómo evolucionaron las métricas de precisión y pérdida a lo largo de las distintas épocas de entrenamiento. Esto se logra mediante el uso de la biblioteca `matplotlib.pyplot` como se puede apreciar en la figura 37.

```
import matplotlib.pyplot as plt

# Graficar la precisión
plt.plot(history.history['accuracy'], label='Entrenamiento')
plt.plot(history.history['val_accuracy'], label='Validación')
plt.xlabel('Época')
plt.ylabel('Precisión')
plt.legend()
plt.show()
```

Figura 37. Visualización del Rendimiento. Autoría propia

1. **Gráfico de Precisión (Accuracy)** En este gráfico se trazan dos curvas como se observa en la figura 38: en una se traza la precisión del entrenamiento del conjunto, mientras que la otra, es para la precisión en la validación del conjunto. La precisión, indica qué porcentaje de predicciones fueron correctas. Por ejemplo, si el modelo clasifica correctamente 80 de cada 100 imágenes, su precisión es del 80 %.

```
# Graficar la pérdida
plt.plot(history.history['loss'], label='Entrenamiento')
plt.plot(history.history['val_loss'], label='Validación')
plt.xlabel('Época')
plt.ylabel('Pérdida')
plt.legend()
plt.show()
```

Figura 38. Precisión en la predicción. Autoría propia

- **history.history['accuracy']** representa la precisión obtenida por el modelo en cada época sobre los datos de entrenamiento.
- **history.history['val_accuracy']** refleja la precisión sobre el conjunto de validación, que son imágenes no vistas por el modelo durante el entrenamiento observado en la figura 39.

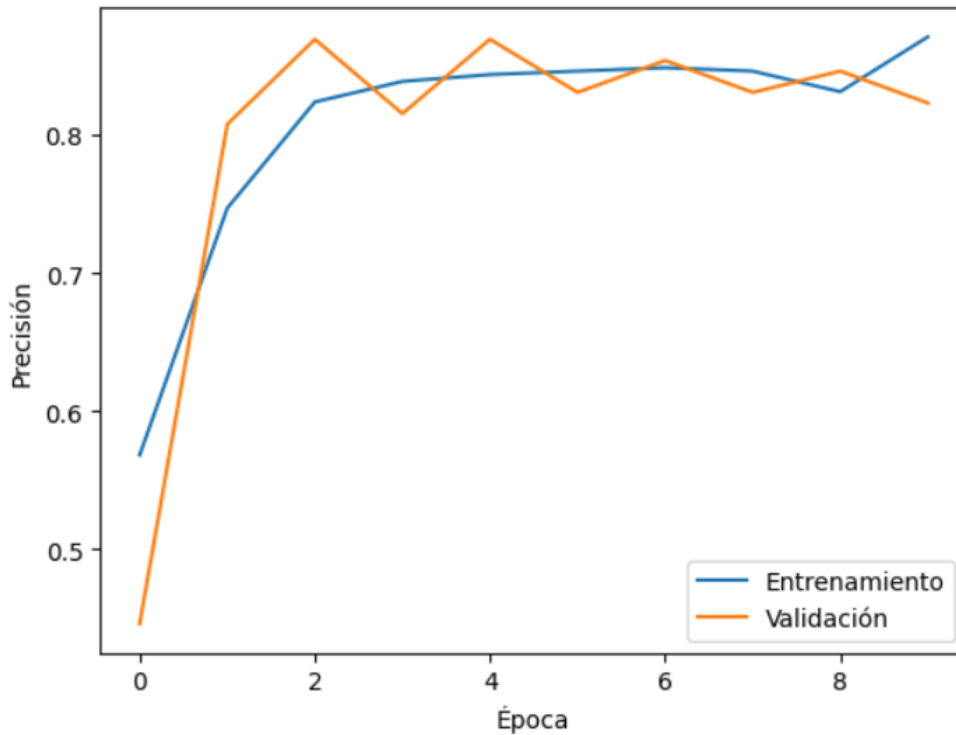


Figura 39. Precisión del modelo. Autoría propia

En el caso del modelo desarrollado para este proyecto, se observa una mejora progresiva en ambas curvas a lo largo de las 10 épocas, con una precisión final superior al 89 % en entrenamiento y más del 82 % en validación, lo cual indica un buen rendimiento.

2. **Gráfico de Pérdida (Loss)** Además de la precisión, otro indicador clave es la función de pérdida. Esta métrica evalúa el error entre las predicciones del modelo y los valores reales. A diferencia de la precisión, mientras menor sea la pérdida, mejor es el modelo mostrados en la figura 40.

- **history.history['loss']** representa la pérdida durante el entrenamiento.
- **history.history['val_loss']** refleja el error sobre el conjunto de validación.

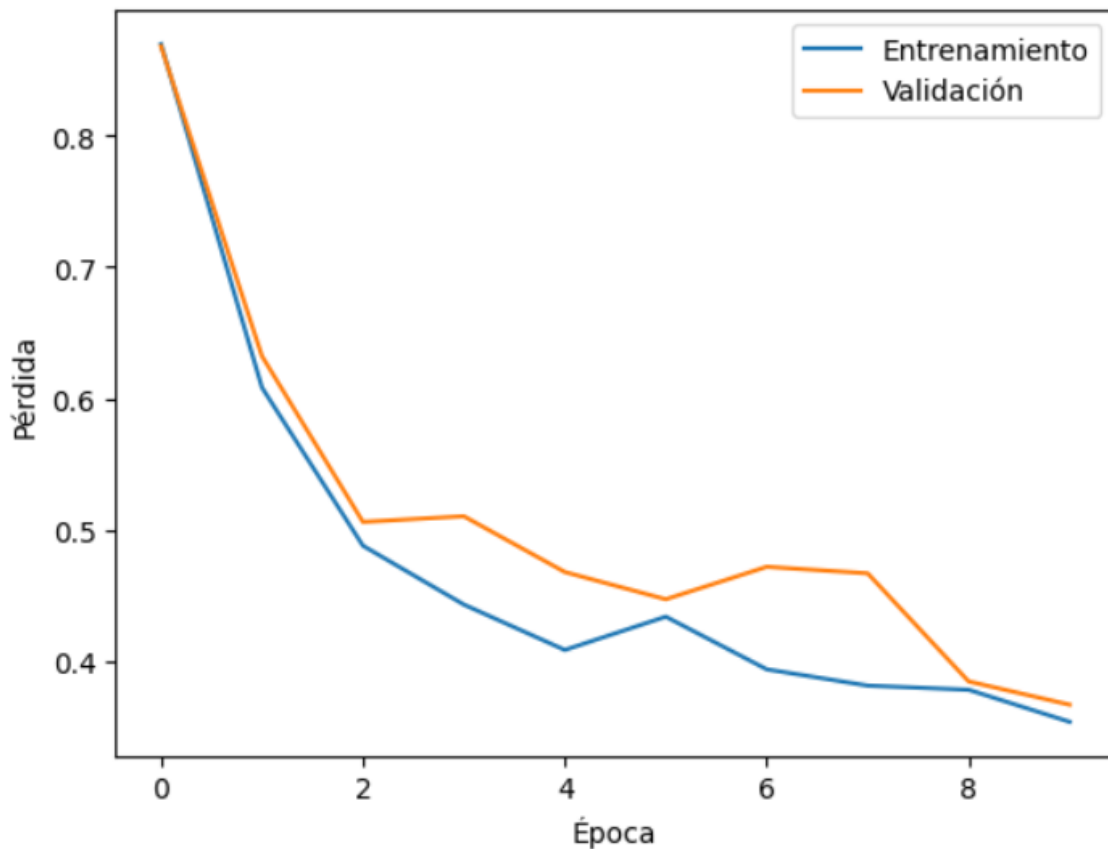


Figura 40. Pérdidas del modelo. Autoría propia

Se puede observar que las curvas de pérdida se mantienen bastante cercanas entre sí, lo que sugiere que el modelo es capaz de generalizar adecuadamente sin caer en sobreajuste, y se adapta bien a nuevos datos.

Ambas curvas son fundamentales para verificar si el modelo está aprendiendo correctamente. Por ejemplo, si la curva de entrenamiento sigue aumentando mientras que la de validación no muestra mejoras o incluso comienza a descender, esto puede señalar la presencia de sobreajuste (overfitting), lo que significa que el modelo está memorizando los datos en lugar de aprender a generalizar.

VI-O. Evaluación del Modelo y Predicción en Tiempo Real

En esta etapa final una vez que el modelo ha sido entrenado y validado satisfactoriamente, es indispensable probar su desempeño en condiciones más similares a las del uso real. Para ello, se toma una imagen aleatoria desde el conjunto de validación y la utiliza para realizar una predicción utilizando el modelo entrenado. Esta simulación representa lo que hará la aplicación móvil final cuando el usuario tome una foto de una mazorca de cacao.

Primero, se importan las bibliotecas necesarias:

1. Importación de Librerías y Carga de Imagen

```
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt
import random
import os
from glob import glob

# Lista de clases (en el mismo orden en que fueron cargadas)
class_names = ['Inmaduro', 'Maduro', 'Sobremaduro']

# Obtener lista de imágenes en valid_path (subcarpetas incluidas)
evaluation_images = glob(os.path.join(valid_path, '*', '*.jpg'))

# Seleccionar aleatoriamente una imagen
random_image_path = random.choice(evaluation_images)

# Cargar y mostrar la imagen
img = image.load_img(random_image_path, target_size=(150, 150))
plt.imshow(img)
plt.axis('off')
plt.title("Imagen del Cacao en su estado de maduración")
plt.show()
```

Figura 41. Evaluación del Modelo. Autoría propia

- **image**, de Keras para cargar y preprocesar la imagen.
- **random**, para seleccionar aleatoriamente una imagen.
- **os** y **glob**, para acceder a las rutas de imágenes.
- **matplotlib.pyplot**, para mostrar gráficamente la imagen.

Luego, se define una lista con las etiquetas de las clases en el mismo orden en que fueron asignadas durante el entrenamiento: 'Inmaduro', 'Maduro' y 'Sobremaduro'.

Se usa **glob()** para recuperar todas las imágenes que se encuentran en las subcarpetas del directorio de validación (**valid_path**). Esto permite simular cómo el sistema manejará imágenes nuevas no vistas por el modelo durante el entrenamiento observado en la figura 41.

Mediante **random.choice()**, se selecciona una imagen aleatoriamente del conjunto, y se carga con **image.load_img()**, redimensionándola a **(150, 150)** píxeles, que es el tamaño de entrada esperado por el modelo. Con **plt.imshow()** se muestra esta imagen, lo que ayuda para depuración como para pruebas visuales en el desarrollo del sistema.

2. Preprocesamiento y Predicción

La imagen cargada es transformada en un arreglo de tipo **NumPy** con **image.img_to_array()**, y expandida una dimensión para simular un “lote” de una sola imagen con **np.expand_dims()**, ya que el modelo espera múltiples imágenes como entrada. Posteriormente, los valores de los píxeles son normalizados dividiendo por 255, igual que durante el entrenamiento. Luego, se llama al método **model.predict()** sobre la imagen procesada. El resultado es un vector de probabilidades, que indica la confianza del modelo en cada una de las

tres clases. Por ejemplo: **[0.05, 0.9, 0.05]** significaría que el modelo está un 90 % seguro de que la mazorca es “Maduro”.

Para determinar la clase final, se utiliza **np.argmax()**, que devuelve el índice con mayor probabilidad, y se lo traduce al nombre de la clase correspondiente usando la lista **class_names**.

Este proceso de inferencia es rápido y eficiente, lo que permite su integración en tiempo real dentro de un aplicativo móvil, permitiendo que un agricultor o técnico pueda obtener al instante el estado de madurez de una mazorca al capturar una imagen desde su celular, usando el código que se muestra en la figura 42.

```
# Preprocesar para predicción
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = img_array / 255.0 # Normalizar como en entrenamiento

# Realizar predicción
prediction = model.predict(img_array)

# Obtener índice de clase predicha
predicted_index = np.argmax(prediction)
predicted_class = class_names[predicted_index]
```

Figura 42. Predicción sobre imagen procesada. Autoría propia

3. Guardado del Modelo

Finalmente, el modelo entrenado se guarda en un archivo **.h5** tal como se observa en la figura 43 utilizando **model.save()**, que puede permitir:

- Reutilizar el modelo en otros entornos como Visual Studio.
- Integrarlo en aplicaciones móviles.
- Evitar tener que entrenarlo nuevamente.

```
print("Predicción (vector de probabilidades):", prediction)
print("Clase predicha:", predicted_class)
model.save('/content/drive/MyDrive/mi_modelo_cacao.h5')
print("Modelo guardado exitosamente en 'mi_modelo_cacao.h5' en Google Drive.")
```

Figura 43. Guardado del modelo. Autoría propia

VII. RESULTADOS

VII-A. Funcionamiento del sistema

La imagen fue procesada mediante re dimensionamiento, normalización y transformación a arreglo NumPy. El modelo devolvió un vector de probabilidades, donde la clase con mayor probabilidad (91.91 %) fue "Sobre maduro". Este resultado fue coherente con las características visuales de la mazorca en la imagen. Finalmente, el modelo fue guardado exitosamente en formato .h5, listo para su posterior uso en un aplicativo web que sirva para soporte agrícola como vemos en la figura 44.



Figura 44. Imagen procesada y evaluada. Autoría propia

VII-B. Evaluación de el sistema de predicción

VII-B1. Resultados de el cacao inmaduro: En la Figura 45 se aprecia una clasificacion de gran precisión de el cacao inmaduro, logrando un 96.15% de precisión media. Esto significa que su reconocimiento es bastante alto. Su desempeño es correcto en diferentes condiciones, asegurando que el cacao inmaduro es reconocible para la IA como vemos en la figura 45.

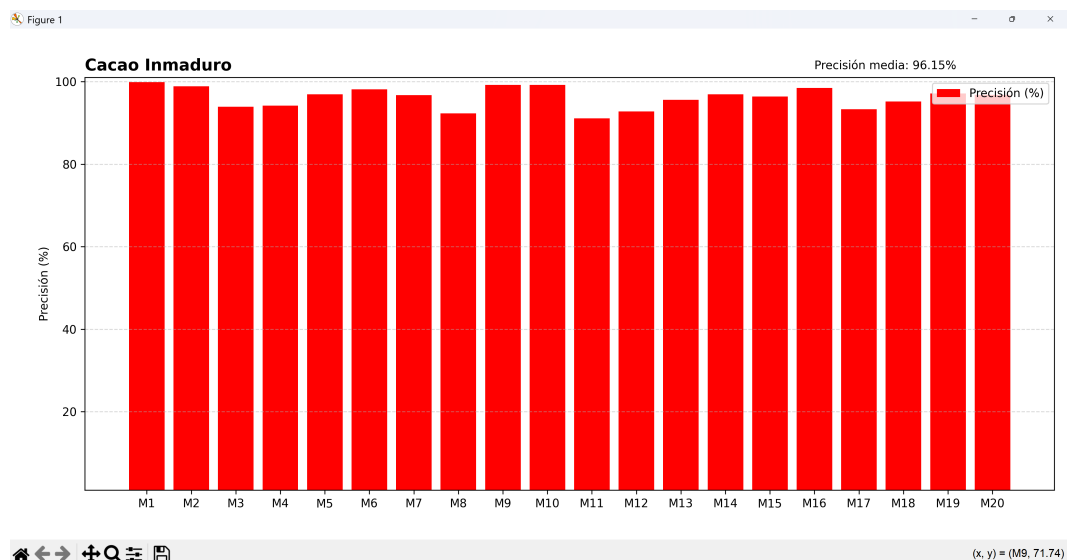


Figura 45. Evaluación cacao inmaduro. Autoría propia

En la figura 46 se muestra la predicción resultante en el aplicativo web diseñado. Usando una imagen de prueba de una mazorca de cacao inmaduro, se muestra una precisión media total del 96.3 %.



Figura 46. Aplicación de visión artificial para el cacao inmaduro.

VII-B2. Resultados de el cacao maduro: En la Figura 47 se aprecia una clasificación de gran precisión de el cacao maduro, logrando un 96.3 % de precisión. lo que es considera un poco menos precisa en comparación del cacao inmaduro. Esto es debido a que no se identifican perfectamente porque el color y la forma varia un poco en este estado, esto podría generar ciertas complicaciones al momento de ser detectada.

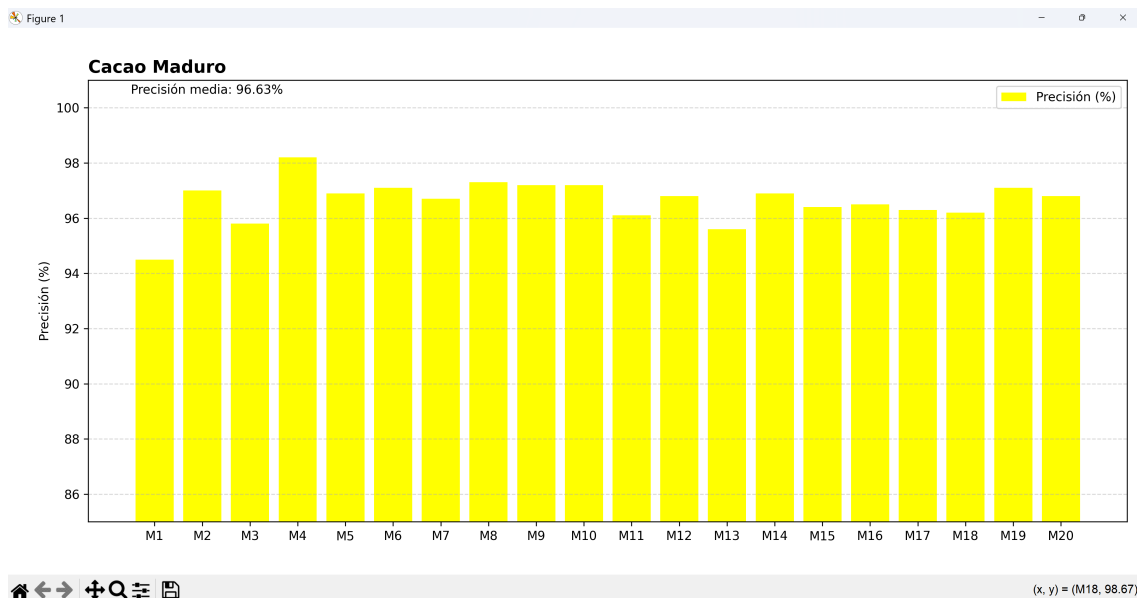


Figura 47. Evaluación cacao maduro. Autoría propia

En la figura 48 se muestra la predicción resultante en el aplicativo web diseñado. Usando una imagen de prueba de una mazorca de cacao maduro, se muestra una precisión total del 96.3 %.



Figura 48. Aplicación de visión artificial para el cacao maduro. Autoría propia

VII-B3. *Resultados de el cacao sobre maduro:* En la Figura 49 se aprecia una clasificación de gran precisión de el cacao sobre maduro, logrando un 96.03 % de aciertos. Lo que es considera bastante aceptable en precisión a comparación del cacao maduro. Esto es debido a que se identifican perfectamente porque el color y la forma varía bastante en este estado, esto se toma como una ventaja pues un cacao sobre maduro es fácilmente reconocible y al momento de ser detectada no presenta dificultad.

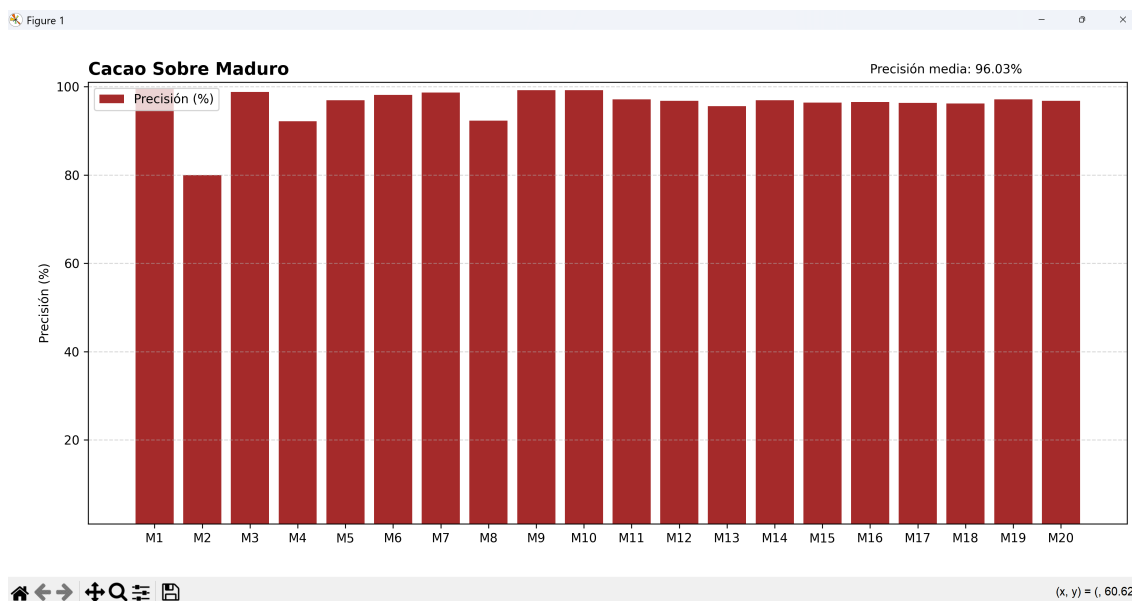


Figura 49. Evaluación cacao sobre maduro. Autoría propia

En la figura 50 se muestra la predicción resultante en el aplicativo web diseñado. Usando una imagen de prueba de una mazorca de cacao inmaduro, se muestra una precisión total del 96.03 %.



Figura 50. Aplicación de visión artificial para el cacao sobre maduro. Autoría propia

Estos resultados fueron trabajados en tiempo real mediante la selección de imágenes aleatorias para el sistema de visión, se tomaron 20 muestras de las cuales una a una fueron insertadas en una gráfica realizada en python con el entorno Visual Studio Code. Como vemos en la figura 51 se puede demostrar que el sistema de identificación de visión artificial funciona de manera correcta indicándonos el estado de maduración en la que se encuentra el cacao y el porcentaje de confiabilidad que tiene el modelo.



Figura 51. Sistema de visión artificial en la pagina web. Autoría propia

El sistema completó con éxito la clasificación de madurez de mazorcas, brindando a los agricultores información precisa, mediante el uso de nuestra red convolucional, diseñada precisamente para dar como resultado predicciones precisas y rápidas. El modelo alcanzó altos valores de precisión (mAP) en el conjunto de prueba, por tanto, se logró una clasificación correcta de la mayoría de las mazorcas analizadas, reduciendo la necesidad de métodos destructivos tradicionales. El sistema opera desde una aplicación web en un dispositivo móvil, donde se sube una imagen previamente capturada y se analizará inmediatamente arrojando el resultado del análisis. En conjunto, el resultado es una aplicación web robusta que identifique automáticamente el nivel de madurez de cada mazorca capturada. Esto permitirá optimizar la cosecha de cacao en Ecuador, especialmente beneficiando a los pequeños productores

Finalmente, los resultados de la clasificación se interpretan para tomar decisiones prácticas, como determinar el momento óptimo de cosecha. Esta información puede ser presentada al usuario a través de una interfaz amigable en un aplicativo web, facilitando su uso en el campo por parte de los agricultores.

VIII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la figura IV.

Tabla IV
CRONOGRAMA

CRONOGRAMA					
		1 MES	2 MES	3 MES	4 MES
OBJETIVO 1					
ACTIVIDAD 1	Desarrollo de la programación	x	X		
ACTIVIDAD 2	Procesamiento	x	X		
ACTIVIDAD 3	Asignación	x	X		
OBJETIVO 2					
ACTIVIDAD 1	Etiquetado	X	x	x	
ACTIVIDAD 2	Entrenamiento		x	x	x
ACTIVIDAD 3	Optimización de algoritmo		x	x	x
ACTIVIDAD 4	Pruebas			X	x
OBJETIVO 3					
ACTIVIDAD 1	Pruebas en campo			X	X
ACTIVIDAD 2	Validación			X	X

IX. PRESUPUESTO

Tabla V
PRESUPUESTO

Nombre del elemento	Descripción	Cantidad	Valor total (USD)
Computadora	De gama alta	1	Propio
Materiales para pruebas	Cacao	1	20
Soportes/Trípodes	Estacion de el equipo	1	20
Telefono	Para prueba	1	Propio
TOTAL COSTOS IMPLEMENTACIÓN			40

X. CONCLUSIONES

1. Se desarrolló un sistema de visión artificial basado en redes neuronales convolucionales, implementando un aplicativo móvil que permite la detección del grado de maduración del cacao mediante un análisis detallado de los patrones visuales de las mazorcas, se logro realizar pruebas con 20 imágenes de prueba para cada estado de maduración en el aplicativo móvil y se lograron llegar a los resultados esperados dando lugar a un indice de maduración con una precisión promedio del 96 % para el estado inmaduro, 96.3 % para el estado maduro y 96.03 % para el estado sobre maduro. Además, el aplicativo móvil brinda una herramienta práctica y accesible para agricultores y productores esto facilita la toma de decisiones en campo y optimizando los procesos de cosecha y pos cosecha.
2. Se diseñó un algoritmo de procesamiento de imágenes para la identificación y clasificación considerando el tamaño y color de las mazorcas de cacao según su etapa de maduración, permitiendo diferenciar de manera automatizada entre frutos inmaduros, maduros y sobremaduros. Este sistema también contribuye a mejorar la precisión en la selección a reducir la subjetividad del análisis humano y optimizar los procesos de control de calidad en la industria cacaotera.
3. Se implementó una aplicación móvil funcional que integra el modelo de clasificación y permite la detección de mazorcas de manera eficiente, proporcionando una interfaz accesible para el usuario y facilitando el procesamiento de imágenes en tiempo real. Esta herramienta tambien optimiza la identificación automática de las etapas de maduración del cacao, y contribuye a la gestión, análisis de la información de forma centralizada, favoreciendo su uso en entornos productivos y académicos.
4. Se validó la exactitud del sistema mediante pruebas de detección de los distintos grados de maduración de las mazorcas de cacao en campo, evaluando su desempeño en condiciones reales de iluminación y variabilidad del entorno. Los resultados obtenidos permitieron conocer el nivel de confiabilidad del modelo en la clasificación automática, demostrando su gran potencial para apoyar procesos agrícolas de selección, mejorando mediante ella la toma de decisiones y garantizando una mayor eficiencia en la industria cacaotera.

XI. RECOMENDACIONES

1. Ampliar la base de datos de entrenamiento alimentando el dataset con imágenes tomadas en diferentes regiones, épocas del año y condiciones climáticas, para mejorar la precisión del modelo y su capacidad de generalización.
2. Explorar futuras funciones del sistema para integrar la detección de otros factores visuales, como enfermedades externas o daños en las mazorcas, o incluir predicciones georreferenciadas para análisis por lote.
3. Difundir el uso entre pequeños y medianos productores mediante una capacitación a los agricultores en el uso de esta herramienta mediante talleres y alianzas con cooperativas, para asegurar su adopción y sostenibilidad tecnológica en el tiempo.
4. Se sugiere establecer un proceso periódico de actualización del modelo, conforme se recolecten más datos, aplicando técnicas como re entrenamiento incremental o aprendizaje activo.

REFERENCIAS

- [1] A. S. M., *Biochemical and pharmacological properties of cacao*. Journal of Agroalimentary Processes y Technologies, 2016.
- [2] B. L. L. J., «Sustainable cacao farming: Challenges and opportunities.,» *134*, 2019.
- [3] J. C. Motamayor, «Geographic and genetic population differentiation of cacao.,» *122*, 2024.
- [4] G. C. W. E., *Digital image processing (4th ed.)*. Pearson, 2018.
- [5] S. R., «Computer vision: Algorithms and applications (2nd ed.,» *102*, 2022.
- [6] R. O., «ImageNet Large Scale Visual Recognition Challenge.1,» *103*, 2015.
- [7] K. Ogata, *Ingeniería de control moderna*. Madrid, España: Pearson Educacion S.A., 2010.
- [8] R. Saltaren, L. Puglisi y J. Sabater, *Robótica aplicada: análisis y diseño de robots paralelos y seriales con MATLAB*. Madrid, España: Dextra Editorial, 2017.
- [9] Ángela Bravo Sánchez, *Control de procesos industriales II*, Accedido: 10-12-2021, 2012. dirección: <https://es.slideshare.net/BlogsalDescubierto/cpi2-clase-4-parte-1-acciones-de-control>.
- [10] J. R. Lajara y J. P. Sebastián, *LabVIEW Entorno gráfico de programación*. Barcelona, España: Marcombo, 2011.
- [11] National Instruments, *LINX de Digilent/LabVIEW Hobbyist Toolkit*, Accedido: 9-01-2022, 2021. dirección: <https://www.ni.com/en-us/support/downloads/tools-network/download.labview-hobbyist-toolkit.html#376574>.
- [12] Naylamp Mechatronics, *Sensor de distancia infrarrojo SHARP GP2Y0A21*, Accedido: 15-02-2022, 2022. dirección: <https://naylampmechatronics.com/sensores-proximidad/203-sensor-de-distancia-infrarrojo-sharp-gp2y0a21.html>.
- [13] R. Dorf y R. Bishop, *Sistemas de control moderno*. Madrid: Pearson Educacion S.A., 2005.
- [14] H. M. M. Rodríguez, *Control de un sistema Bola Balancín*. Valladolid, España: Escuela de Ingenierías Industriales, 2019.
- [15] E. Rivas, E. Cruz y P. Vela, *Diseño de un sistema electrónico y electroneumático basado en visión artificial para la selección y clasificación de barras de chocolate blanco*. Piura, Perú: Facultad de Ingeniería Industrial, 2021.
- [16] L. Greenwood, *Agrocuriosidades: El cultivo y aprovechamiento del cacao*, 2016.
- [17] B. C. A. S. G. E., «Estudio socio-económico del cultivo de cacao (Theobroma cacao L.) en la parroquia Febres Cordero, Cantón Babahoyo Los Ríos-Ecuador.,» *Observatorio de la Economía Latinoamericana*, 2019.
- [18] M. L., *Maduración y recolección de la maduración del cacao*, 2023.
- [19] Proamazonia, *ORÍGENES DEL CACAO ECUATORIANO - Proamazonia*, 2020.
- [20] Sandra, *Producción de cacao en Ecuador y su impacto económico*.
- [21] P. G. Francisco Sotomayor, «Sistema de Visión Artificial para el Análisis de Imágenes de Cultivo basado en Texturas Orientadas,» *Revista Politécnica*, 2014. dirección: https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/104.
- [22] B. P. Ana Garcia, «La cadena de producción del Cacao en Ecuador: Resiliencia en los diferentes actores de la producción,» *Novasinergia*, 2021. dirección: <https://doi.org/10.37135/ns.01.08.10>.
- [23] V. Quinde Rosales, R. Bucaram Leverone, M. Bucaram Leverone y M. Bueno Quiñonez, *Factores productivos de la producción de cacao nacional de la Provincia del Guayas*.

- [24] I. J. F. Carvajal, A. R. I. Cedeño, N. M. S. Castro, D. B. V. Marcillo y R. L. V. Almeida, «La producción de cacao en la provincia de Manabí: Un enfoque de cadena de valor,» *Polo del Conocimiento: Revista científico-profesional*, vol. 8, n.º 5, págs. 166-185, 2023.
- [25] V. Torres y D. Nayelhy, «Análisis del proceso poscosecha y agroindustrial de la producción de cacao Theobroma cacao L., El Carmen, Manabí,» Tesis doct., 2023.
- [26] H. R. Claudia Russo, «Visión artificial aplicada en Agricultura de Precisión,» *ITT ARTICLES*, 2018. dirección: <http://repositorio.unnoba.edu.ar/xmlui/handle/23601/662>.
- [27] A. P. Rosero Montalvo, «Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura,» *Enfoque UTE*, 2021. dirección: <https://doi.org/10.29019/enfoqueute.v8n1.121>.
- [28] A. Ibarra, «Análisis de la cadena de cacao en la provincia de los ríos, Ecuador,» 84, 2019.
- [29] C. C. Cuevas Castañeda, «La cadena de producción del Cacao en Ecuador: Resiliencia en los diferentes actores de la producción,» *Universidad Nacional Abierta y a Distancia UNAD*, 2019. dirección: <https://repository.unad.edu.co/handle/10596/27853>.
- [30] M. D. Aranda, «Aprendizaje Automático aplicado a la calidad del desarrollo en la ganadería de precisión,» *AJEA - Actas de Jornadas y Eventos Académicos de UTN*, 2022. dirección: <https://doi.org/10.33414/ajea.1040.2022>.
- [31] Q. L. Xintong Ji, «A Machine Learning-Based Method for Pig Weight Estimation and the PIGRGB-Weight Dataset. Agriculture,» *MEDPI*, 2025. dirección: <https://doi.org/10.3390/agriculture15080814>.
- [32] Z. W. S. S. E. Chan, «ASAS-NANP SYMPOSIUM: Applications of machine learning for livestock body weight prediction from digital images,» *OXFORD ACADEMIC*, 2021. dirección: <https://doi.org/10.1093/jas/skab022>.
- [33] A. Sojasingarayar, «Faster R-CNN vs YOLO vs SSD — Object Detection Algorithms,» *MEDIUM*, 2022. dirección: <https://medium.com/ibm-data-ai/faster-r-cnn-vs-yolo-vs-ssd-object-detection-algorithms-18badb0e02dc>.
- [34] keviczky L., «Object detection using YOLOv5 from scratch with Python — Computer Vision,» *Object detection*, 2014.
- [35] S. Haykin, *Neural Networks and Learning Machines*, 3.^a ed. Upper Saddle River, NJ: Pearson, 2009.
- [36] S. D. S. S., «In Google cloud platform for data science: A crash course on big data, machine learning, and data analytics services,» *Berkeley, CA*, 2023.
- [37] F. P., «The effects of public labelling.,» *Criminology*, 1977.
- [38] M. Corporation, *Visual studio in the new era*, 2025.
- [39] K. A., «Deep learning in agriculture,» *Agriculture CBB*, 1990.
- [40] T. L. S. J., «Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques,» *IGI Global Scientific Publishing*, 2010.

ANEXO A

CÓDIGO USADO EN GOOGLE COLAB PARA LA REALIZACIÓN DEL MODELO DEL SISTEMA DE RECONOCIMIENTO DE VISIÓN ARTIFICIAL

```
# General imports
import numpy as np
import pandas as pd
import os
from glob import glob
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

↻ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

train_path = '/content/drive/MyDrive/train'
valid_path = '/content/drive/MyDrive/valid'

# Leer el número de imágenes en las carpetas de entrenamiento y prueba
num_train_images = len(os.listdir(train_path))
num_test_images = len(os.listdir(valid_path))
print("Número de imágenes en la carpeta de entrenamiento:", num_train_images)
print("Número de imágenes en la carpeta de prueba:", num_test_images)

↻ Número de imágenes en la carpeta de entrenamiento: 3
Número de imágenes en la carpeta de prueba: 3

train_images = glob(os.path.join(train_path, '/content/drive/MyDrive/train/Inmaduro', '/content/drive/MyDrive/train/Sobremaduro', '/content/valid_images = glob(os.path.join(valid_path, '/content/drive/MyDrive/valid/Inmaduro', '/content/drive/MyDrive/train/Sobremaduro', '/content/
#print opcional para saber que si me reconoce las carpetas
print("Número de imágenes en la carpeta de entrenamiento:", len(train_images))
print("Número de imágenes en la carpeta de prueba:", len(valid_images))

↻ Número de imágenes en la carpeta de entrenamiento: 1
Número de imágenes en la carpeta de prueba: 1

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
# Configuración
batch_size = 32
target_size = (150, 150) # Puedes ajustar el tamaño según tu modelo

# Generador para entrenamiento
train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical', # + Cambiado de 'binary' a 'categorical'
    shuffle=True # Mezcla aleatoria para mejor entrenamiento
)

# Generador para validación
valid_datagen = ImageDataGenerator(rescale=1./255)

valid_generator = valid_datagen.flow_from_directory(
    valid_path,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False # No es necesario mezclar en validación
)

# Mostrar clases detectadas
print("Clases detectadas automáticamente:")
print(train_generator.class_indices)

↻ Found 403 images belonging to 3 classes.
Found 130 images belonging to 3 classes.
Clases detectadas automáticamente:
{'Inmaduro': 0, 'Maduro': 1, 'Sobremaduro': 2}

# Definir el modelo
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(pool_size=(2, 2)),
```

```

Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(pool_size=(2, 2)),

Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D(pool_size=(2, 2)),

Flatten(),

Dense(128, activation='relu'),
Dropout(0.5),

Dense(3, activation='softmax') # 3 clases de madurez del cacao
])

# Compilar el modelo
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy', # Porque usamos etiquetas one-hot
    metrics=['accuracy']
)

# Mostrar resumen del modelo
model.summary()

```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_4 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_5 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_4 (Dense)	(None, 128)	4,735,104
dropout_1 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 3)	387

Total params: 4,828,739 (18.42 MB)
Trainable params: 4,828,739 (18.42 MB)

```

# Entrenar el modelo
history = model.fit(
    train_generator,
    epochs=10, # Puedes ajustar según tu tiempo/datos
    validation_data=valid_generator
)

```

```

Epoch 1/10
13/13 ----- 24s 2s/step - accuracy: 0.5336 - loss: 0.9407 - val_accuracy: 0.4462 - val_loss: 0.8684
Epoch 2/10
13/13 ----- 23s 2s/step - accuracy: 0.6819 - loss: 0.6623 - val_accuracy: 0.8077 - val_loss: 0.6321
Epoch 3/10
13/13 ----- 21s 2s/step - accuracy: 0.8475 - loss: 0.4565 - val_accuracy: 0.8692 - val_loss: 0.5057
Epoch 4/10
13/13 ----- 22s 2s/step - accuracy: 0.8335 - loss: 0.4555 - val_accuracy: 0.8154 - val_loss: 0.5101
Epoch 5/10
13/13 ----- 21s 2s/step - accuracy: 0.8570 - loss: 0.3832 - val_accuracy: 0.8692 - val_loss: 0.4675
Epoch 6/10
13/13 ----- 24s 2s/step - accuracy: 0.8576 - loss: 0.4259 - val_accuracy: 0.8308 - val_loss: 0.4468
Epoch 7/10
13/13 ----- 21s 2s/step - accuracy: 0.8385 - loss: 0.4038 - val_accuracy: 0.8538 - val_loss: 0.4715
Epoch 8/10
13/13 ----- 22s 2s/step - accuracy: 0.8583 - loss: 0.3867 - val_accuracy: 0.8308 - val_loss: 0.4665
Epoch 9/10
13/13 ----- 21s 2s/step - accuracy: 0.8457 - loss: 0.3676 - val_accuracy: 0.8462 - val_loss: 0.3841
Epoch 10/10
13/13 ----- 23s 2s/step - accuracy: 0.8931 - loss: 0.3305 - val_accuracy: 0.8231 - val_loss: 0.3664

```

```
import matplotlib.pyplot as plt
```

```

# Graficar la precisión
plt.plot(history.history['accuracy'], label='Entrenamiento')
plt.plot(history.history['val_accuracy'], label='Validación')
plt.xlabel('Época')
plt.ylabel('Precisión')

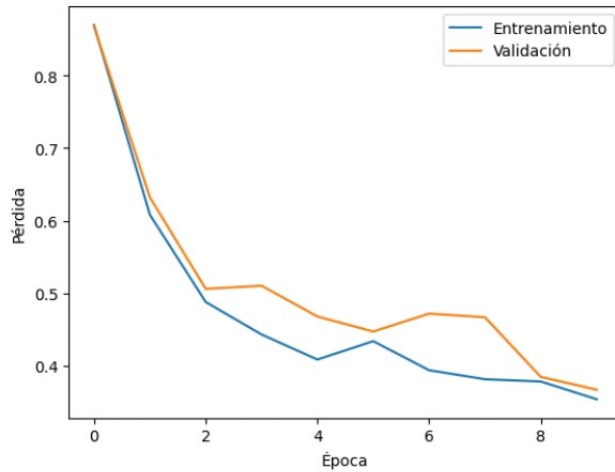
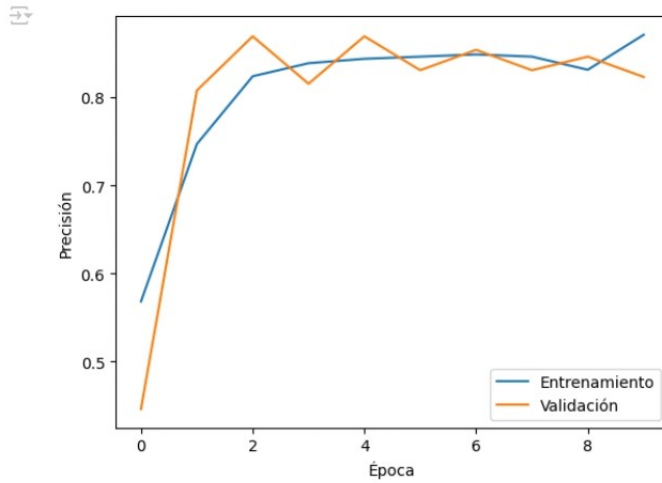
```

```

plt.legend()
plt.show()

# Graficar la pérdida
plt.plot(history.history['loss'], label='Entrenamiento')
plt.plot(history.history['val_loss'], label='Validación')
plt.xlabel('Época')
plt.ylabel('Pérdida')
plt.legend()
plt.show()

```



```

from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt
import random
import os
from glob import glob

# Lista de clases (en el mismo orden en que fueron cargadas)
class_names = ['Inmaduro', 'Maduro', 'Sobremaduro']

# Obtener lista de imágenes en valid_path (subcarpetas incluidas)
evaluation_images = glob(os.path.join(valid_path, '*', '*.*'))

# Seleccionar aleatoriamente una imagen
random_image_path = random.choice(evaluation_images)

# Cargar y mostrar la imagen
img = image.load_img(random_image_path, target_size=(150, 150))
plt.imshow(img)
plt.axis('off')
plt.title("Imagen del Cacao en su estado de maduración")
plt.show()

# Preprocesar para predicción

```

```

img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = img_array / 255.0 # Normalizar como en entrenamiento

# Realizar predicción
prediction = model.predict(img_array)

# Obtener índice de clase predicha
predicted_index = np.argmax(prediction)
predicted_class = class_names[predicted_index]

print("Predicción (vector de probabilidades):", prediction)
print("Clase predicha:", predicted_class)
model.save('/content/drive/MyDrive/mi_modelo_cacao.h5')
print("Modelo guardado exitosamente en 'mi_modelo_cacao.h5' en Google Drive.")

```

Imagen del Cacao en su estado de maduración



```

1/1 ----- 0s 104ms/step
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is
Predicción (vector de probabilidades): [[0.0237987 0.19708896 0.7791124 ]]
Clase predicha: Sobremaduro
Modelo guardado exitosamente en 'mi_modelo_cacao.h5' en Google Drive.

```

ANEXO B

CÓDIGO DE PYTHON USADO EN VISUAL STUDIO CODE PARA EL RECONOCIMIENTO DE VISIÓN ARTIFICIAL

```

CODIGO_CACAO_LIVE.py x  <> Index.html  <> prediccion.html  Grafica.py  <> Base.html  ▶ ▢ ...
CODIGO_CACAO_LIVE.py > ...
1  from flask import Flask, render_template, request, jsonify
2  from keras.models import load_model
3  from keras.preprocessing import image
4  import numpy as np
5  import os
6  from PIL import Image
7
8  app = Flask(__name__)
9
10 # Ruta del modelo .h5
11 model_path = os.path.join('C:\\Users\\pchac\\Desktop\\PROYECTO_CACO_LIVE', 'mi_modelo_cacao.h5')
12
13 # Cargar el modelo
14 try:
15     model = load_model(model_path)
16     print("✅ Modelo cargado correctamente.")
17 except Exception as e:
18     print("❌ Error al cargar el modelo:", str(e))
19
20 @app.route('/')
21 def home():
22     return render_template('index.html')
23

```

```
CODIGO_CACAO_LIVE.py × <> Index.html <> prediccion.html Grafica.py <> Base.html
CODIGO_CACAO_LIVE.py > ...
23
24 @app.route('/prediccion')
25 def mostrar_prediccion():
26     return render_template('prediccion.html')
27
28 @app.route('/predict', methods=['POST'])
29 def predict():
30     if 'image' not in request.files:
31         return '✘ No se encontró la imagen en la solicitud.', 400
32
33     uploaded_file = request.files['image']
34
35     if uploaded_file.filename == '':
36         return '✘ No se seleccionó ninguna imagen.', 400
37
38     try:
39         # Guardar imagen temporalmente
40         image_path = 'uploaded_image.jpg'
41         uploaded_file.save(image_path)
42
43         # Preprocesar la imagen
44         img = image.load_img(image_path, target_size=(150, 150))
45         img_array = image.img_to_array(img)
46         img_array = np.expand_dims(img_array, axis=0)
```

```
CODIGO_CACAO_LIVE.py × <> Index.html <> prediccion.html Grafica.py <> Base.html
CODIGO_CACAO_LIVE.py > ...
29 def predict():
49     # Realizar predicción
50     prediction = model.predict(img_array)
51
52     # Multiclase
53     class_names = ['Cacao Inmaduro', 'Cacao Maduro', 'Cacao Sobremaduro']
54     predicted_index = np.argmax(prediction)
55     predicted_class = class_names[predicted_index]
56     confidence = float(prediction[0][predicted_index]) * 100
57
58     # Limpiar
59     if os.path.exists(image_path):
60         os.remove(image_path)
61
62     return f'Predicción: <strong>{predicted_class}</strong><br>Confianza: {confidence:.2f}%'
63
64 except Exception as e:
65     return f'✘ Error al procesar la imagen: {str(e)}', 500
66
67
68 if __name__ == '__main__':
69     app.run(host='127.0.0.1', port=5000, debug=True)
```

ANEXO C

CÓDIGO HTML PARA LA PAGINA DE INICIO DE LA PAGINA WEB

```
CODIGO_CACAO_LIVE.py  Index.html x  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>CacaoPredict - Predicción de Maduración</title>
7  <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&family=Open+Sans:wght@400;700" rel="stylesheet">
8  <style>
9  /* Estilos Globales y Reset */
10 html {
11     scroll-behavior: smooth;
12 }
13
14 body {
15     margin: 0;
16     font-family: 'Open Sans', sans-serif;
17     background-color: #f9f1e7;
18     color: #3b2c21;
19 }
20
```

```
CODIGO_CACAO_LIVE.py  Index.html x  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2  <html lang="es">
484 <body>
486 <nav>
487     <div class="nav-left">
488         
489         <h1>CacaoPredict</h1> </div>
490     <button class="menu-toggle" onclick="toggleMenu()">☰</button>
491     <div class="nav-right" id="menu">
492         <a href="/">Inicio</a>
493         <a href="/prediccion">Predicción</a>
494         <a href="#acerca-de">Acerca de</a> <a href="#como-funciona">Cómo Funciona</a> <a href="#info-cacao">Información</a>
495         <a href="#contacto">Contacto</a>
496     </div>
497 </nav>
498
499 <div class="hero-section">
500     
501     <div class="hero-overlay"></div> <div class="hero-content">
502         <h2>CacaoPredict: La Inteligencia que Tu Cultivo Necesita</h2>
503         <p>Optimiza tu cosecha de cacao con nuestra predicción precisa del estado de maduración, potenciada por visión por computador e IA.</p>
504         <div class="button-group">
505             <a href="/prediccion" class="button">Iniciar Predicción</a> <a href="#acerca-de" class="button">Conoce la Metodología</a>
506         </div>
507     </div>
508 </div>
```

```

CODIGO_CACAO_LIVE.py  Index.html X  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2 <html lang="es">
484 <body>
510 <section id="acerca-de">
511 <h2>¿Por qué CacaoPredict?</h2>
512 <div class="problem-solution-grid">
513 <div class="grid-item">
514 <h3>El Problema Actual</h3>
515 <p>Identificar el momento óptimo de maduración del cacao es un desafío constante para los agricultores. Los métodos tradiciona
516 <ul>
517 <li>Subjetivos y dependientes de la experiencia.</li>
518 <li>Ineficientes y propensos a errores.</li>
519 <li>Causan pérdidas por cosecha prematura o tardía.</li>
520 <li>Dificultan la estandarización de la calidad.</li>
521 </ul>
522 </div>
523 <div class="grid-item">
524 <h3>Nuestra Solución Innovadora</h3>
525 <p>CacaoPredict transforma la toma de decisiones en tu cultivo. Nuestra herramienta te permite:</p>
526 <ul>
527 <li>Predicción precisa y objetiva de la maduración.</li>
528 <li>Maximizar la calidad y el rendimiento del cacao.</li>
529 <li>Reducir pérdidas y optimizar el tiempo de cosecha.</li>
530 <li>Tomar decisiones basadas en datos concretos.</li>

```

```

CODIGO_CACAO_LIVE.py  Index.html X  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2 <html lang="es">
484 <body>
510 <section id="acerca-de">
512 <div class="problem-solution-grid">
533 </div>
534 </section>
535
536 <section id="como-funciona">
537 <h2>Así Funciona CacaoPredict</h2>
538 <div class="steps-container">
539 <div class="step-item">
540 <div class="icon"><img alt="Icono de cámara" /></div> <h3>1. Sube tu Imagen</h3>
541 <p>Fotografía tu vaina de cacao y cárgala en nuestra plataforma de manera sencilla.</p>
542 </div>
543 <div class="step-item">
544 <div class="icon"><img alt="Icono de cerebro" /></div> <h3>2. Análisis Inteligente</h3>
545 <p>Nuestros algoritmos de Visión por Computador e Inteligencia Artificial procesan la imagen.</p>
546 </div>
547 <div class="step-item">
548 <div class="icon"><img alt="Icono de checkmark" /></div> <h3>3. Obtén la Predicción</h3>
549 <p>Recibe al instante el estado de maduración exacto y nuestras recomendaciones para la cosecha.</p>
550 </div>
551 </div>

```

```

CODIGO_CACAO_LIVE.py  Index.html X  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2 <html lang="es">
484 <body>
555 <section id="info-cacao">
556 <h2>Información Esencial sobre el Cacao</h2>
557 <p>El cacao (Theobroma cacao L.) es más que un fruto; es un pilar fundamental de la economía agrícola en Ecuador y el mundo. Su calidad
558
559 <div class="image-gallery">
560 
561 </div>
562
563 <div class="maduracion-detalle">
564 <h3>Importancia de la Maduración</h3>
565 <p>El punto de maduración ideal asegura el desarrollo óptimo de los precursores de sabor, la concentración de nutrientes y la facil
566
567 <h3>Estados Clave de Maduración del Cacao</h3>
568 <p><strong>Inmaduro:</strong> El fruto presenta una coloración predominantemente verde o amarilla clara, dependiendo de la variedad
569 <p><strong>Maduro (Óptimo para Cosecha):</strong> El fruto adquiere una coloración intensa y uniforme (amarillo, naranja, rojo o mo
570 <p><strong>Sobremaduro:</strong> El fruto comienza a deteriorarse, a menudo con manchas oscuras, pudrición o incluso la germinación
571 </div>
572 </section>
573

```

```

CODIGO_CACAO_LIVE.py  Index.html X  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2 <html lang="es">
484 <body>
573
574 <section id="contacto">
575 <div class="footer-content">
576 <h2>Contáctanos</h2>
577 <p>Si tienes preguntas o deseas saber más sobre CacaoPredict y nuestro proyecto de tesis:</p>
578 <p><strong>Autor:</strong> Miguel Narea</p>
579 <p><strong>Correo:</strong> <a href="mailto:mnareat@est.ups.edu.ec">mnareat@est.ups.edu.ec</a></p>
580 <p><strong>Teléfono:</strong> +593 996331833</p>
581 <div class="social-links">
582 <a href="https://www.linkedin.com/in/miguel-angel-narea-torres-517828245/" target="_blank" aria-label="LinkedIn de Miguel Narea">
583 <a href="https://github.com/miguelnarea03-oss" target="_blank" aria-label="GitHub del Proyecto">Gh</a>
584 </div>
585 <p style="margin-top: 20px;">&copy; 2025 CacaoPredict. Desarrollado como Proyecto de Tesis por Miguel Narea. Todos los derechos res
586 </div>
587 </section>
588

```

```

CODIGO_CACAO_LIVE.py  Index.html X  prediccion.html  Grafica.py  Base.html
templates > Index.html > html
2 <html lang="es">
484 <body>
574 <section id="contacto">
587 </section>
588
589 <script>
590 // JavaScript para el menú hamburguesa
591 function toggleMenu() {
592     document.getElementById("menu").classList.toggle("show");
593 }
594
595 // Opcional: Cerrar menú si se hace clic fuera o se redimensiona
596 window.addEventListener('resize', function() {
597     if (window.innerWidth > 768) {
598         document.getElementById("menu").classList.remove("show");
599     }
600 });
601 </script>
602
603 </body>
604 </html>

```

ANEXO D

CÓDIGO HTML PARA LA PAGINA DE PREDICCIÓN DE LA PAGINA WEB

```

CODIGO_CACAO_LIVE.py  Index.html  prediccion.html X  Grafica.py  Base.html
templates > prediccion.html > html > head > style > body
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1" />
6 <title>Predicción de Maduración de Cacao</title>
7 <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Open+Sans&display=swap" rel="stylesheet">
8 <style>
9 /* Estilos CSS incrustados para la página de predicción */
10 body {
11     font-family: 'Open Sans', sans-serif;
12     background: linear-gradient(135deg, #6a4a2b 0%, #a9744f 100%); /* Fondo de gradiente marrón */
13     margin: 0;
14     padding: 0;
15     min-height: 100vh;
16     color: #f3e9dc; /* Color de texto claro */
17     background-image: url('https://www.transparenttextures.com/patterns/pw-pattern.png'); /* Textura sutil de fondo */
18     background-repeat: repeat;
19     background-size: contain;
20     text-align: center;
21 }
22

```

```

CODIGO_CACAO_LIVE.py  Index.html  predicción.html X  Grafica.py  Base.html
templates > <> predicción.html > html > head > style > body
2 <html lang="es">
213 </head>
214 <body>
215
216 <nav class="navbar">
217   <div class="navbar-content">
218     <h1 class="logo"> CacaoPredict</h1>
219     <ul class="nav-links">
220       <li><a href="{{ url_for('home') }}">Inicio</a></li>
221     </ul>
222   </div>
223 </nav>
224
225 <div class="container">
226   <h1>Predicción de Maduración de Cacao</h1>
227   <form id="predictionForm" action="/predict" method="post" enctype="multipart/form-data" onsubmit="submitForm(event)">
228     <label for="image">Seleccionar imagen</label>
229     <input type="file" name="image" id="image" accept="image/*" onchange="previewImage(this)" />
230     <img id="preview" alt="Imagen seleccionada" />
231     <button type="submit">Predecir Estado</button>
232   </form>
233   <div id="prediction"></div>
234

```

```

CODIGO_CACAO_LIVE.py  Index.html  predicción.html X  Grafica.py  Base.html
templates > <> predicción.html > html > head > style > body
2 <html lang="es">
214 <body>
236
237 <script>
238   // Función para previsualizar la imagen seleccionada por el usuario
239   function previewImage(input) {
240     const preview = document.getElementById('preview');
241     if (input.files && input.files[0]) {
242       const reader = new FileReader();
243       reader.onload = function (e) {
244         preview.src = e.target.result;
245         preview.style.display = 'block'; // Muestra la imagen
246       };
247       reader.readAsDataURL(input.files[0]);
248     } else {
249       preview.src = '';
250       preview.style.display = 'none'; // Oculta la imagen si no hay archivo
251     }
252   }
253
254   // Función para enviar el formulario usando Fetch API (AJAX)
255   function submitForm(event) {
256     event.preventDefault(); // Evita el envío tradicional del formulario
257     const form = document.getElementById('predictionForm');

```

```
CODIGO_CACAO_LIVE.py  Index.html  predicción.html X  Grafica.py  Base.html
templates > predicción.html > html > head > style > body
2  <html lang="es">
214 <body>
237 <script>
255   function submitForm(event) {
257       const form = document.getElementById('predictionForm');
258       const formData = new FormData(form);
259
260       // Envía los datos al endpoint '/predict' de tu servidor Flask
261       fetch('/predict', {
262         method: 'POST',
263         body: formData,
264       })
265       .then(response => response.text()) // Espera una respuesta de texto
266       .then(data => {
267         // Muestra la respuesta del servidor en el div 'prediction'
268         document.getElementById('prediction').innerHTML = data;
269       })
270       .catch(error => {
271         // Manejo de errores
272         console.error('Error:', error);
273         document.getElementById('prediction').innerHTML = 'Error al realizar la predicción. Por favor, inténtalo de nuevo.';
274       });
275   }
```

ANEXO E
EJEMPLO DE DATASETS USADOS PARA EL ENTRENAMIENTO DEL MODELO TANTO
INMADURO, MADURO Y SOBRE MADURO RESPECTIVAMENTE

