

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Clasificación de defectos en manzanas usando aprendizaje profundo con imágenes del espectro visible y del espectro infrarrojo cercano.

TECH-340

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES**

Presentado por:

Omar Mauricio Coello Camacho

Moisés Agustín Coronel Unda

GUAYAQUIL - ECUADOR

Año: 2023

## **Dedicatoria**

---

El presente proyecto lo dedico a mi familia, pero de manera especial a mi mamá por su apoyo y enseñanzas a lo largo de mi formación académica. Sus valiosos consejos han sido fundamentales para alcanzar mis objetivos profesionales. Gracias por inculcarme valores de perseverancia y superación.

### **Moisés Coronel**

A mí mismo, por los días de incansable trabajo, las noches de inquebrantable determinación y los momentos de autorreflexión que han dado forma a este trabajo. Este trabajo es el fruto de mi dedicación, mi pasión y mi búsqueda constante de conocimiento. Agradezco a aquellos que me han acompañado de cerca y a aquellos que han dejado una huella en mi camino.

### **Omar Coello**

## Agradecimientos

---

Nuestro más sincero agradecimiento al Ph.D. Boris Vintimilla y al Ing. Dario Carpio por su paciencia, orientación y guía durante este proyecto. Su experiencia y conocimiento fueron fundamentales para superar los desafíos y lograr resultados exitosos.

También queremos agradecer al Ing. Jonathan Paillacho por su valiosa ayuda y apoyo técnico en el desarrollo del proyecto. Su contribución fue de gran valor para el éxito de este proyecto.

## Declaración Expresa

---

Nosotros Omar Mauricio Coello Camacho y Moisés Agustín Coronel acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

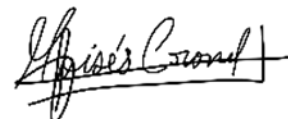
En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 13 de febrero del 2024.



---

Omar Coello



---

Moisés Coronel

# Evaluadores

---

Lucia Marisol Villacrés, PhD.

Profesor de Materia

---

Boris Vintimilla Burgos, PhD.

Tutor de proyecto

## RESUMEN

Este estudio aborda la clasificación de defectos en manzanas, la cual es crucial para mitigar pérdidas económicas y optimizar la cadena de suministro de alimentos. Se propone un enfoque innovador que integra imágenes del espectro visible y del infrarrojo cercano (NIR) para mejorar la precisión y eficiencia en la detección, a diferencia de investigaciones previas que se enfocaban en un solo espectro. La metodología involucra el uso de redes neuronales convolucionales feedforward y siamesas para procesar imágenes RGB-NIR capturadas en paralelo, siguiendo los pasos de adquisición, preprocesamiento, entrenamiento del modelo clasificador y evaluación del rendimiento.

Los resultados demostraron que la detección de defectos utilizando el espectro NIR reveló detalles no visibles en el espectro visible e incluso que su uso fue ligeramente superior. El modelo seleccionado alcanzó una precisión del 98.80% en validación. En conclusión, el estudio destacó el potencial de mejorar la detección al capturar imágenes con rangos más amplios de filtros NIR, permitiendo una mejor segmentación para el entrenamiento de redes siamesas y aumentar la capacidad de identificar y clasificar defectos en manzanas. Este novedoso enfoque de integrar múltiples espectros en la clasificación de defectos agrícolas muestra prometedoras aplicaciones en la mejora de procesos de selección y clasificación.

**Palabras Clave:** Espectro cercano al infrarrojo (NIR), Aprendizaje profundo, Procesamiento de imágenes, Clasificación de defectos, Redes siamesas.

## ABSTRACT

This study addresses the classification of defects in apples, which is crucial for mitigating economic losses and optimizing the food supply chain. An innovative approach is proposed that integrates images from the visible spectrum and near-infrared (NIR) spectrum to improve accuracy and efficiency in defect detection, unlike previous research that focused on a single spectrum. The methodology involves using feedforward and siamese convolutional neural networks to process RGB-NIR images captured in parallel, following the steps of acquisition, preprocessing, training the classification model, and evaluating performance.

The results demonstrated that defect detection using the NIR spectrum revealed details not visible in the visible spectrum and even showed slightly superior performance. The selected model achieved an accuracy of 98.80% on validation. In conclusion, the study highlighted the potential to improve detection by capturing images with broader ranges of NIR filters, enabling better segmentation for siamese network training and increasing the ability to identify and classify defects in apples. This novel approach of integrating multiple spectra in agricultural defect classification shows promising applications in improving selection and classification processes.

**Keywords:** Near-infrared (NIR) spectrum, Deep learning, Image processing, Defect classification, Siamese networks.

# ÍNDICE GENERAL

Resumen .....	VI
Abstract.....	VII
Índice general .....	VIII
Abreviaturas .....	XI
Simbología.....	XII
Índice de figuras .....	XIII
Índice de tablas .....	XIV
Capítulo 1 .....	15
1.    Introducción .....	16
1.1    Descripción del Problema.....	18
1.2    Justificación del Problema.....	19
1.3    Objetivos.....	20
1.3.1    Objetivo general.....	20
1.3.2    Objetivos específicos .....	20
1.4    Marco teórico.....	21
1.4.1    Normativas y estándares de calidad de las manzanas .....	21
1.4.2    Normativas de etiquetado y empaque .....	21
1.4.3    Defectos en frutas .....	22
1.4.4    Tecnologías de imágenes en la agricultura .....	25
1.4.5    Espectro electromagnético y sus aplicaciones en agricultura .....	26



1.4.6	Adquisición de imágenes .....	27
1.4.7	Detección de defectos en frutas por medio de visión por computador usando arquitecturas CNN.....	29
1.4.8	Estudios de casos relevantes .....	30
Capítulo 2 .....		33
2.	Metodología. ....	34
2.1	Diseño de la solución.....	34
2.2	Módulo 1: Creación del dataset .....	35
2.2.1	Adquisición de imágenes .....	35
2.2.2	Etiquetado de dataset .....	37
2.2.3	Preprocesamiento de imágenes .....	38
2.2.4	Data augmentation .....	39
2.3	Módulo 2: Entrenamiento de modelos.....	40
2.3.1	Arquitectura de red tipo Feedforward.....	41
2.3.2	Arquitectura de red siamesa.....	43
2.4	Módulo 3: Evaluación de modelos .....	45
Capítulo 3 .....		47
3.	ANÁLISIS Y RESULTADOS.....	48
3.1	Creación del conjunto de imágenes .....	48
3.1.1	Dataset de imágenes RGB y NIR.....	48
3.1.2	Dataset de imágenes siluetas en el espectro RGB y NIR.....	51
3.1.3	Data augmentation .....	54

3.2	Arquitectura de los modelos .....	55
3.2.1	Justificación de selección de modelos .....	55
3.2.2	Hiperparámetros utilizados para el entrenamiento .....	56
3.3	Entrenamiento de modelos .....	58
3.3.1	Proceso de entrenamiento de modelos <i>feedforward</i> .....	58
3.3.2	Proceso de entrenamiento de modelos siamesa .....	59
3.3.3	Evaluación de métricas de rendimiento durante el entrenamiento y validación 61	
3.4	Selección del modelo final .....	62
3.4.1	Criterios utilizados para seleccionar el modelo final .....	63
3.5	Evaluación del modelo .....	64
3.5.1	Análisis de métricas de evaluación .....	64
3.5.2	Matriz de confusión y análisis de errores .....	66
3.6	Análisis de Errores.....	67
3.7	Interpretación de resultados.....	68
Capítulo 4	.....	69
4.	Conclusiones Y Recomendaciones .....	70
4.1	Conclusiones.....	70
4.2	Recomendaciones .....	72
Referencias	.....	74
APÉNDICES	.....	77

## **ABREVIATURAS**

ESPOL Escuela Superior Politécnica del Litoral

CiDiS Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales

FAO Food and Agriculture Organization

NIR Near Infra Red

RGB Red Green Blue

CNN Convolutional Neural Network

SAM Segmentation Anything Model

## **SIMBOLOGÍA**

nm      Nanometros

db      Decibelio

## ÍNDICE DE FIGURAS

Figura 1.1 Tipos comunes de defectos en productos hortícolas .....	22
Figura 1.2 Manzana con magulladura .....	24
Figura 1.3 Manzana con manchas .....	24
Figura 1.4 Manzana en estado de podredumbre .....	25
Figura 1.5 Espectro electromagnético .....	26
Figura 2.1 Módulos de desarrollo de la solución .....	35
Figura 2.2 Curva de respuesta espectral de cámara NIR.....	36
Figura 2.3 Imagen RGB y NIR con defecto golpe .....	37
Figura 2.4 Imágenes RGB y NIR con su respectiva silueta .....	38
Figura 2.5 Arquitecturas Feedforwards propuestas.....	42
Figura 2.6 Arquitecturas Siamesas propuestas .....	44
Figura 3.1 Silueta resultantes generadas para las imágenes RGB y NIR.....	54
Figura 3.2 Curva de accuracy de modelo feedforward MobilenetV1 NIR .....	63
Figura 3.3 Matriz de confusión MobileNetV1 con imágenes NIR reajustadas.....	66

## ÍNDICE DE TABLAS

Tabla 1.1 Comparación de modelos de CNN para detectar defectos en frutas [12,13] ....	30
Tabla 3.1 Cantidad de imágenes RGB y NIR con defecto de golpes.....	49
Tabla 3.2 Cantidad de imágenes RGB y NIR con defectos de manchas y pudrición. ....	50
Tabla 3.3 Accuracy de modelos feedforward.....	64

# Capítulo 1

# 1. INTRODUCCIÓN

El control de calidad de las frutas es un proceso fundamental para garantizar la excelencia en la industria agrícola de los países productores de estos alimentos. El control de calidad de las frutas convencionalmente se realiza manualmente mediante la inspección visual de las personas. Sin embargo, este proceso tradicional presenta limitaciones significativas, como altos costos laborales, baja velocidad e imprecisión, por mencionar unos cuantos inconvenientes, lo que puede ocasionar pérdidas económicas considerables[1].

Existen estándares internacionales desarrollados por organismos como la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) que evalúan tres parámetros clave para determinar la calidad de las frutas: madurez valorada por la proporción de azúcares y ácidos que impacta características como color, textura y sabor; deformaciones incluyendo desviaciones de forma por daños mecánicos en crecimiento/recolección; y defectos como lesiones o enfermedades que afectan la apariencia[2]. De acuerdo con la evaluación de estos tres factores, la calidad se clasifica en: Extra Class (sin defectos apreciables), First Class (leves defectos, pero buena calidad comercial) y Second Class (aceptable para consumo, aunque con defectos más notorios y menor valor). Esta clasificación en tres grados define estándares mínimos de calidad e inocuidad para exportación e internacionalización de frutas de manera global.

En este proyecto se propone desarrollar un sistema de detección de defectos en frutas usando técnicas de aprendizaje profundo con imágenes del espectro visible y del espectro infrarrojo cercano (NIR). El espectro visible se refiere a la radiación electromagnética que es perceptible para el ojo humano, y que se extiende desde los 380 nanómetros (nm) de longitud de



onda de la luz violeta hasta los 700 nm de la luz roja. Esta radiación constituye solo una pequeña parte del espectro electromagnético total.

El espectro infrarrojo cercano (NIR, por sus siglas en inglés Near Infrared), se ubica justo después del espectro visible, desde los 700 nm hasta los 2500 nm. En esta región del espectro, la energía de la radiación es suficientemente alta como para excitar los enlaces de vibración de moléculas como el agua, proteínas y aceites en los alimentos. Dado que diferentes materiales absorben la radiación NIR de forma característica, analizando el espectro de absorción es posible realizar análisis cuantitativos no destructivos de las propiedades y composición química de los alimentos, como humedad, aceites, proteínas, entre otros. Esto hace que la espectroscopía NIR sea una técnica muy valiosa para aplicaciones de control de calidad y grado en la industria alimentaria.

Es crucial identificar los defectos de forma temprana en las frutas, ya que esto no solo asegura su calidad, sino también su valor nutricional. Evitar que las frutas no aptas para el consumo lleguen al mercado es esencial para mantener la confianza de los consumidores. Aunque los métodos de inspección visual tradicional han sido ampliamente utilizados, el uso de técnicas de visión por computador y aprendizaje profundo ha demostrado un gran potencial para mejorar la detección de defectos.

Este proyecto se enfoca en la evaluación de al menos 4 modelos de aprendizaje profundo para la detección de defectos en manzanas utilizando imágenes capturadas en el espectro visible e infrarrojo. Para lograr este objetivo, se emplearán los modelos: ResNet50, DenseNet121, MobilNetv1, VGG19. Estos modelos se entrenarán con datos representativos de ejemplos de defectos comunes en la fruta.

La fruta que se usará para el presente estudio es la manzana y sus defectos pueden variar ampliamente, desde manchas, deformaciones, decoloraciones, hasta daños físicos y pudriciones. Identificar estos defectos de manera precisa y rápida es esencial para garantizar un proceso de control de calidad eficiente y confiable.

## **1.1 Descripción del Problema**

La detección manual de defectos en las frutas representa un alto costo para este tipo de industria. Se estima que entre el 20-30% de la mano de obra en la cosecha y postcosecha corresponde a la inspección y selección de las frutas y/o vegetales [3]. Considerando que el promedio salarial de los trabajadores agrícolas en la región varía entre los US\$100 y US\$500 mensuales [3], los costos laborales asociados son significativos.

Por otro lado, se estima que al menos un 14% de la producción se pierde debido a daños y defectos no detectados antes de llegar al nivel minorista [4], generando pérdidas económicas millonarias para los países productores. Los principales defectos se deben a plagas, enfermedades, maduración irregular, lesiones mecánicas y variaciones genéticas [4]. También hay deficiencias internas como hongos y bacterias, que no son visibles a simple vista y pasan desapercibidas en los controles manuales habituales [5].

La detección temprana y precisa de defectos en frutas representa un reto tecnológico y económico para América Latina, dada su importante contribución al sector agroalimentario y a la economía de la región. Por tanto, se requieren soluciones innovadoras basadas en visión computacional y aprendizaje profundo, que permitan superar las limitaciones de los métodos manuales.

## 1.2 Justificación del Problema

La detección temprana y clasificación automatizada de defectos en frutas como manzanas es fundamental para la industria agrícola y agroalimentaria de América Latina y otros países productores [1]. La implementación de un sistema de detección automatizado permitiría separar de manera rápida y precisa las frutas defectuosas de las saludables durante la cosecha, empaque y distribución. Esto es crucial para evitar significativas pérdidas económicas debido a manzanas de baja calidad o al rechazo de mercados.

Las pérdidas postcosecha a causa de defectos representan un gran desafío. Según reportes, en la última década representaron alrededor del 30% de la producción total anual en la región, lo que equivale a miles de millones de dólares en pérdidas [2]. Un sistema automatizado de detección temprana permitiría reducir sustancialmente estas cifras, beneficiando tanto a productores como economías nacionales. De esta forma, la implementación de la tecnología propuesta sería muy valiosa para asegurar la calidad y trazabilidad de frutas desde origen hasta consumidor final, fortaleciendo la competitividad del sector [6].

La automatización también reduce considerablemente los costos laborales asociados con la inspección visual manual, que requiere de una alta mano de obra. Se estima que se necesitan un promedio de 20 trabajadores para revisar un contenedor de 40 pies de frutas [3], lo que implica salarios significativos. Adicionalmente, la inspección manual es un proceso lento y tiene un alto margen de error. La implementación de tecnologías digitales como la automatización de procesos, la robótica y el machine learning permiten que las empresas atiendan las necesidades de servicio de sus unidades de negocio con celeridad y eficiencia [7]. Asimismo, estas tecnologías posibilitan que los recursos humanos se enfoquen en tareas más especializadas y estratégicas para el negocio.

El eficiente uso de la tecnología permite la detección temprana y precisa de plagas y enfermedades facilitando el monitoreo y control oportuno de estas, preservando así la calidad e inocuidad de los cultivos. Esto se traduce en una disminución en el uso de agroquímicos, contribuyendo a la sostenibilidad del sector agrícola.

### **1.3 Objetivos**

#### **1.3.1 Objetivo general**

Desarrollar un sistema de detección automatizado de defectos en frutas utilizando tecnologías de aprendizaje profundo y procesamiento de imágenes, con un enfoque en la integración de imágenes del espectro visible y del espectro infrarrojo cercano (NIR), con el fin de mejorar la calidad de los productos agrícolas y optimizar la cadena de suministro de alimentos al proporcionar una detección precisa y eficiente de defectos en las manzanas.

#### **1.3.2 Objetivos específicos**

1. Generar un conjunto de imágenes de manzanas reales que representen una variedad de defectos incluyendo manchas, magulladuras y podredumbre en el espectro visible e infrarrojo cercano (NIR).
2. Entrenar al menos 4 modelos de aprendizaje profundo en dos tipos de arquitecturas: feedforward y siamesas, utilizando el conjunto de datos generado previamente para identificar y clasificar los defectos en las manzanas.
3. Evaluar el rendimiento de los modelos de aprendizaje profundo en la clasificación de defectos en frutas, utilizando métricas de precisión, sensibilidad y especificidad para medir su capacidad de detección y clasificación.

## **1.4 Marco teórico**

### **1.4.1 Normativas y estándares de calidad de las manzanas**

En Ecuador, la calidad de las manzanas se rige por normativas y estándares específicos establecidos por autoridades competentes y organizaciones agrícolas.

La Norma Técnica Ecuatoriana (NTE) establece los requisitos mínimos de calidad para las manzanas producidas y comercializadas en Ecuador. Puede incluir parámetros como tamaño, color, firmeza, contenido de azúcares y ausencia de defectos graves. La presencia de defectos visibles, como daños mecánicos, manchas o deformidades, puede afectar la clasificación y categorización de las manzanas según su calidad [8].

### **1.4.2 Normativas de etiquetado y empaque**

Las normativas del Instituto Ecuatoriano de Normalización (INEN) pueden abordar requisitos específicos para el etiquetado y el empaque de frutas, incluyendo información sobre origen, clasificación, fecha de envasado y manejo postcosecha. El empaque debe proteger las frutas de daños durante el transporte y garantizar su frescura al llegar al consumidor.

Cumplir con estas normativas es esencial para los productores y exportadores de frutas en Ecuador, ya que asegura la calidad, seguridad y competitividad de los productos en los mercados nacionales e internacionales. El conocimiento y cumplimiento de estas normativas son aspectos clave en la gestión de la calidad y la comercialización exitosa de las frutas producidas en el país [8].

### 1.4.3 Defectos en frutas

Los defectos de frutas son imperfecciones o anomalías que se presentan en la apariencia, textura, sabor o calidad de las frutas. Los defectos pueden causarse por diversos factores, bióticos y abióticos, y pueden tener un impacto significativo en la comercialización, consumo y rentabilidad de la fruta [4].

Para identificar el tipo de imperfección vinculado a una fruta o hortaliza, es esencial clasificar las fallas en categorías específicas que caractericen una causa al manifestar defectos en los alimentos. La Figura 1.1 proporciona una descripción detallada de estas categorías, las cuales se diferencian según las prácticas de cosecha, la manipulación posterior y las condiciones de almacenamiento, como se menciona en [18].



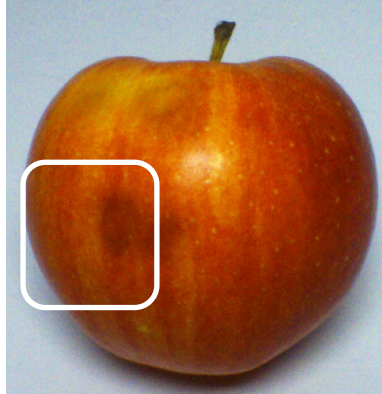
**Figura 1.1 Tipos comunes de defectos en productos hortícolas**

Los 4 tipos de defectos más comunes en frutas son:

- **Daños mecánicos:** Golpes, magulladuras, cortes, abolladuras causadas durante la cosecha, manipulación, transporte o almacenamiento. Son los defectos no patológicos más frecuentes [3].
- **Desórdenes patológicos:** Son defectos causados principalmente por hongos como *Penicillium* y *Botrytis*, que generan manchas y podredumbres. También bacterias y virus causantes de diferentes enfermedades [3].
- **Desórdenes morfológicos:** Se refieren a irregularidades en la forma y estructura de la fruta, generalmente causadas por factores genéticos, ambientales o de manejo [3].
- **Desórdenes fisiológicos:** Originados por factores ambientales como estrés térmico, hídrico o nutricional. Pueden manifestarse como decoloración, rajado, deshidratación, maduración irregular [3].
- **Defectos internos:** Estos defectos no son evidentes externamente, ya que se encuentran dentro de la fruta [3].

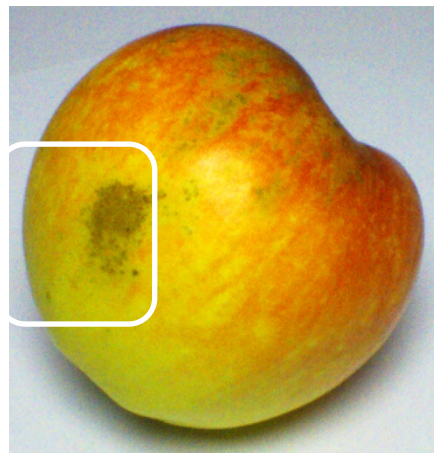
Para este proyecto se tomará en consideración estos tres defectos particulares para su evaluación:

- **Magulladuras:** Las magulladuras son daños mecánicos en la epidermis de la fruta, generados normalmente por golpes durante la cosecha, manipulación o transporte. Pueden presentarse de diversos tamaños y formas irregulares, dejando zonas amoratadas o enrojecidas en la superficie. Representan una entrada potencial para hongos y bacterias, acelerando el deterioro del producto. Su detección automática permite seleccionar frutas aptas para la comercialización.



**Figura 1.2 Manzana con magulladura**

- **Manchas:** Las manchas en la fruta pueden estar causadas por diferentes hongos, como *Penicillium* y *Alternaria*. Se manifiestan generalmente en forma de áreas descoloridas, irregulares, con tonos cremosos, grises o parduzcos. Algunas inicialmente son pequeñas, pero luego pueden expandirse rápidamente si no son controladas. Su detección oportuna posibilita aplicar tratamientos fungicidas que eviten mayores pérdidas en la cosecha.

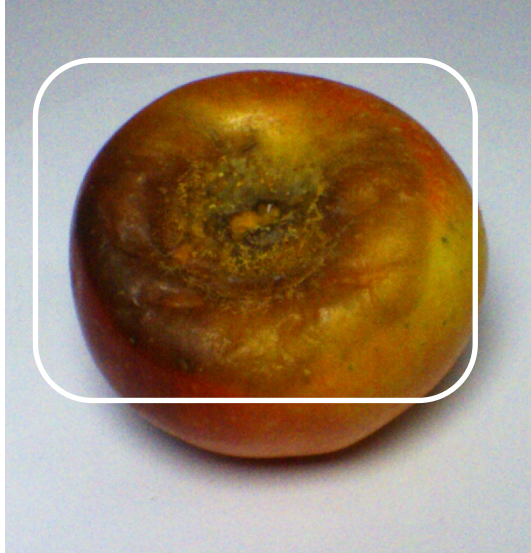


**Figura 1.3 Manzana con manchas**

- **Podredumbre:** La podredumbre es causada por hongos o bacterias que logran infectar el tejido interno de la fruta, aprovechando cualquier lesión. Puede observarse como zonas blandas, de coloración anómala y mal olor, generalmente creciendo desde la zona dañada hacia el interior. Es uno de los mayores problemas postcosecha ya que en etapas



avanzadas toda la fruta se pierde, por lo que métodos de detección temprana permiten desechar solo las partes afectadas.



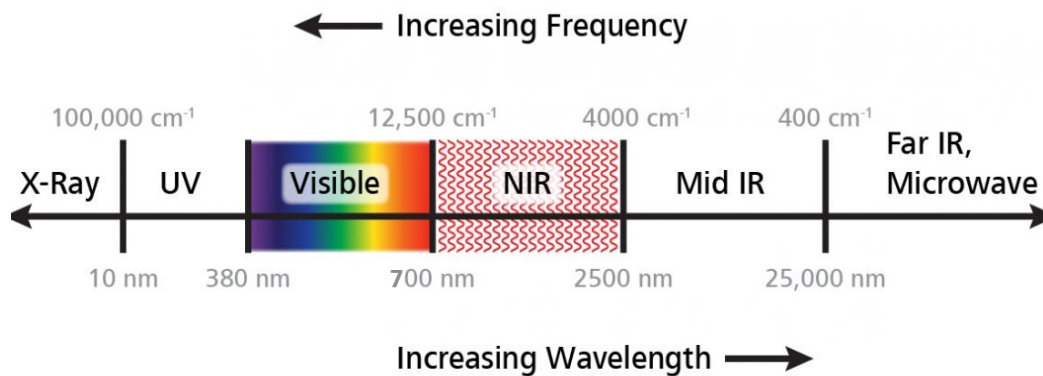
**Figura 1.4 Manzana en estado de podredumbre**

#### **1.4.4 Tecnologías de imágenes en la agricultura**

La aplicación de tecnologías de imágenes, como las imágenes del espectro visible y multiespectrales, ha demostrado ser eficaz en la monitorización de la salud de los cultivos [7]. Las imágenes del espectro visible proporcionan información sobre las características superficiales de las frutas, mientras que las imágenes multiespectrales permiten la detección de patrones y condiciones invisibles a simple vista. Además, las imágenes multiespectrales también permiten analizar a detalle las condiciones internas de la fruta, como magulladuras y putrefacciones.

### 1.4.5 Espectro electromagnético y sus aplicaciones en agricultura

El espectro electromagnético abarca un amplio rango de longitudes de onda, desde las ondas de radio hasta los rayos gamma, tal como se visualiza en la Figura 1.5 . Para aplicar en agricultura, se presta atención a rangos espectrales en particular, por su utilidad para detectar defectos en frutas con facilidad. Entre los rangos más relevantes se encuentran.



**Figura 1.5 Espectro electromagnético**

El rango visible (380-700 nm) el cual incluye la luz visible para el ojo humano. Las imágenes en el espectro visible son fundamentales para evaluar la apariencia superficial de las frutas, detectar cambios en el color y examinar defectos que afectan la pigmentación, como manchas y descoloración.

Según un estudio previamente realizado [8] en el que se efectuaron pruebas con varios filtros de paso de banda en varias subregiones del espectro NIR con imágenes adquiridas desde una cámara NIR, se concluyó que una subregión del infrarrojo cercano ubicada entre 700 a 1300 nm, es útil para evaluar el estado de las plantas y frutas debido a que ciertos pigmentos vegetales tienen respuestas específicas en esta banda. Puede proporcionar información sobre la

composición química y la estructura celular, lo que es beneficioso para la detección temprana de problemas de salud en las frutas.

Una de las ventajas del uso de rangos fuera del espectro visible es que permite detectar cambios no visibles a simple vista, como daños internos, variaciones en la composición química y problemas de salud en las frutas. Además, al analizar regiones espectrales específicas, es posible identificar defectos en las frutas antes de que sean visibles externamente. Esto es crucial para la gestión proactiva de la calidad y la prevención de pérdidas.

El espectro electromagnético permite un análisis cuantitativo de las propiedades de las frutas, como la concentración de pigmentos, la humedad y otros componentes, lo que proporciona información detallada sobre la calidad de las frutas [8]. Adicionalmente, algunas bandas espectrales son menos afectadas por condiciones de iluminación, lo que proporciona estabilidad en la detección independientemente de factores ambientales.

#### **1.4.6 Adquisición de imágenes**

Las imágenes capturadas, después de pasar por el preprocesamiento y la segmentación, se utilizan como entrada para la CNN. Cada imagen representa una instancia de fruta con sus posibles defectos.

#### **Cámaras RGB en detección de defectos de frutas**

Las cámaras RGB son una opción común en sistemas de visión para la industria frutícola, ya que permiten identificar defectos basados en diferencias de color. Sin embargo, solo proporcionan información de 3 bandas espectrales, por lo que defectos sutiles pueden pasar desapercibidos. Aun así, su rápida adquisición de imágenes las hace ideales para detección en línea de defectos evidentes como quemaduras o daños mecánicos [16].

### **Cámaras infrarrojas para detección de defectos de frutas**

Las cámaras infrarrojas capturan imágenes en el espectro infrarrojo cercano, lo que proporciona una visión detallada de las frutas mediante la captura de sus firmas espectrales en bandas específicas de longitud de onda infrarroja. Estas firmas espectrales únicas permiten identificar con precisión defectos, incluso incipientes o relacionados con factores internos como la presencia de hongos. Aunque su procesamiento puede requerir un tiempo ligeramente mayor, las cámaras infrarrojas son altamente efectivas en aplicaciones de investigación y control de calidad donde se necesita una detección precisa [14].

### **Especificaciones de computadora utilizada para entrenamiento**

La computadora utilizada para el entrenamiento de todos los modelos fue un PC armado con procesador Intel(R) Core (TM) i9-10920X CPU @ 3.50GHz, 32 GB de RAM instalada, HDD con 2 TB de almacenamiento, tarjeta gráfica NVIDIA GeForce RTX3090 24 GB GDDR6X, sistema operativo de 64 bits, Windows 10 Pro.

Se utilizó la versión 2.10.1 de TensorFlow y la 10.2 CUDA Version para el entrenamiento de los modelos con arquitectura feedforward y Siamesa.

#### **1.4.7 Detección de defectos en frutas por medio de visión por computador usando arquitecturas CNN.**

La detección precisa de defectos en frutas es esencial para garantizar la calidad y la comercialización exitosa de los productos agrícolas. En este contexto, la visión por computadora es poderosa, y las arquitecturas de redes neuronales convolucionales (CNN) han demostrado ser eficaces en la identificación automática de patrones en imágenes agrícolas.

En este subtema, se explorará la aplicación específica de arquitecturas CNN para la detección de defectos en frutas, centrándose en manzanas en el contexto agrícola. La visión por computadora se convierte en un aliado estratégico al emplear técnicas de aprendizaje profundo para analizar imágenes capturadas de forma no invasiva, proporcionando una evaluación rápida y precisa de la calidad de las frutas.

Las redes neuronales convolucionales (CNN) nos permitirán identificar defectos en frutas. Para el entrenamiento de las CNN se empleen las arquitecturas que hayan demostrado los mayores niveles de precisión al detectar defectos en estas frutas, ya que se busca obtener los mejores resultados posible[24]. Aunque existen estudios previos que se enfocaron en analizar defectos en frutas usando redes neuronales convolucionales, para este estudio se considerarán los trabajos evaluados con manzanas, ya que es la fruta que se analizarán en este proyecto [23]. Del mismo modo, serán relevantes aquellos estudios que hayan evaluado defectos utilizando espectros de imagen comparables al que se empleará en esta investigación.

### 1.4.8 Estudios de casos relevantes

Fruta	Defectos	Modelo	Desempeño	
			RGB	NIR
Manzana [13]	Golpes	VGG19	MCC - 0.969	MCC - 0.967
		ResNet50	MCC - 0.970	MCC - 0.970
		DenseNet121	MCC - 0.978	MCC - 0.956
Manzana [14]	Golpes, costras, podredumbre, puntos negros	InceptionV3	Accuracy - 93.40%	
		MobileNetV2	Accuracy - 97.50%	
		DenseNet121	Accuracy - 94.50%	
		VGG16	Accuracy - 76.20%	
Mango [14]	Golpes, costras, podredumbre, puntos negros	InceptionV3	Accuracy - 93.40%	
		MobileNetV2	Accuracy - 97.50%	
		DenseNet121	Accuracy - 94.50%	
		VGG16	Accuracy - 76.20%	
Limón [15]	Golpes	ResNet50	Accuracy - 90.47%	
		DenseNet121	Accuracy - 85.71%	
		ShuffleNet	Accuracy - 80.95%	
		MobileNetV1	Accuracy - 73.80%	

**Tabla 1.1 Comparación de modelos de CNN para detectar defectos en frutas [12,13]**

Los estudios [13], [15] que utilizaron imágenes hiperespectrales demostraron la viabilidad de detectar magulladuras en frutas como manzanas y limones en el espectro del

infrarrojo cercano. Estos estudios identificaron las longitudes de onda óptimas en dicho espectro, las cuales contenían más información para distinguir tejidos sanos de lesionados. En particular, se encontró que las longitudes de 750nm, 850nm y 960nm eran ideales para diferenciar magulladuras.

Los resultados del estudio [15] indican que la arquitectura ResNet-50 logró la mejor precisión en la clasificación de imágenes hiperespectrales de limones con daños celulares, con una tasa de clasificación del 98,33%. Los otros modelos de CNN también obtuvieron resultados prometedores, con tasas de clasificación superiores al 95%. Mientras que en el estudio [13] se logró un alto nivel de precisión en la clasificación de manzanas sanas y con golpes. La arquitectura Densenet-121 obtuvo un promedio de *coeficiente de correlación de Matthews (MCC)* de 97,8 % en la validación cruzada de 5 pliegues, lo que indica que casi todas las manzanas se clasificaron correctamente. Además, se menciona que ResNet50 también se utilizó en uno de los experimentos y obtuvo un promedio de MCC de 97%.

En el estudio [14] se muestra que MobileNetV2 fue el modelo CNN que alcanzó los mayores porcentajes de precisión tanto para manzanas como para mangos, superando al resto de arquitecturas en ambos casos. Los resultados sugieren que este modelo es el más adecuado para la detección automática de defectos en dichas frutas. DenseNet121 e InceptionV3 tuvieron un buen rendimiento, mientras que VGG16 fue el peor clasificador entre los evaluados.

De acuerdo con el análisis realizado sobre estudios previos de detección de defectos en frutas mediante técnicas de visión por computadora, se ha demostrado que algunos modelos de redes neuronales convolucionales han arrojado excelentes resultados. En particular, los estudios [13] y [14] reportan que, para la inspección de manzanas, las arquitecturas ResNet-50, DenseNet-121 y MobileNet-V1 lograron altos niveles de precisión superior al 95%. Asimismo,

el estudio [14] mostró que, para mangos, MobileNet-V1 también fue el modelo más efectivo, obteniendo altos porcentajes de precisión. Dada la similitud morfológica y composición entre limones, manzanas y mangos, para el presente estudio de detección de defectos en manzanas se seleccionarán los modelos ResNet50, DenseNet121, MobileNetV1 y VGG19, debido a los buenos resultados previos al identificar correctamente defectos en frutas similares con imágenes de espectro similares.



## **Capítulo 2**

## **2. METODOLOGÍA.**

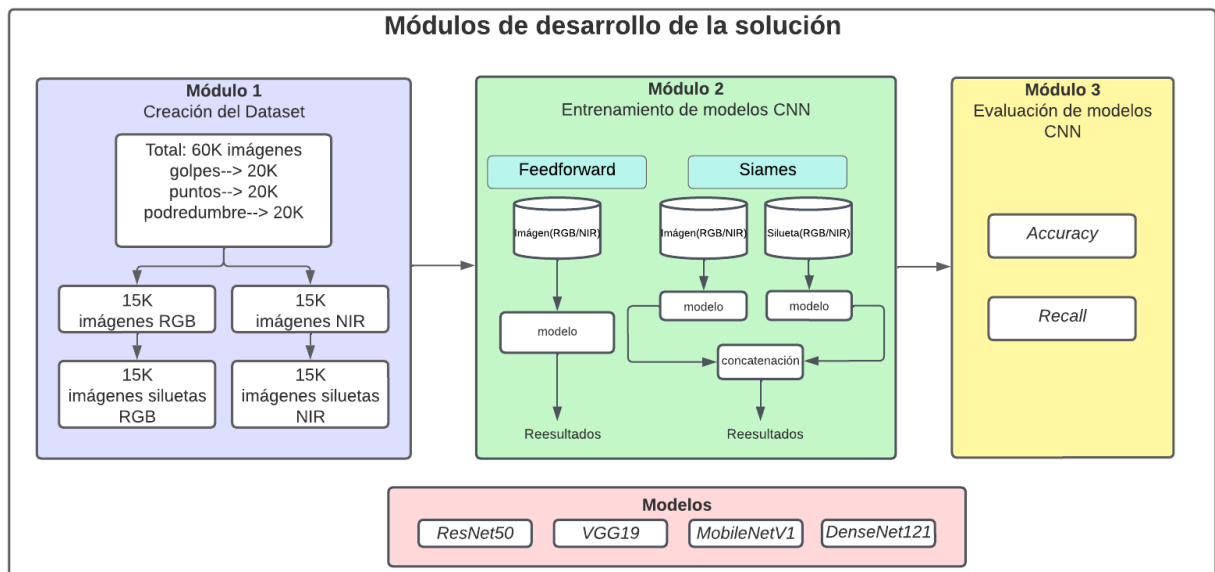
En este capítulo se describe la metodología utilizada para la generación de las bases de datos de las imágenes necesarias para este proyecto, para esto, se explica el proceso de adquisición de imágenes, lo cual incluyó imágenes tanto en el espectro visible como del espectro infrarrojo cercano (NIR). Asimismo, se detalla el procedimiento usado para el entrenamiento de las redes neuronales convolucionales (CNN) que se evaluaron en la detección de defectos de manzanas. Además, se brinda una breve descripción de cada CNN seleccionada para la tarea de detección de defectos, que se sometieron a entrenamiento y optimización para evaluar su desempeño.

### **2.1 Diseño de la solución**

La solución propuesta se diseñó con una arquitectura modular que consta de tres componentes o módulos principales, tal como se ilustra en la Figura 2.1. El módulo 1 de esta solución corresponde a la creación del dataset de imágenes que se usaron para el proyecto, el cual constará de un total de 60 mil imágenes de manzanas con 3 tipos de defectos, esto es: manchas, golpes y podredumbre. Este dataset está compuesto por imágenes del espectro visible RGB y sus correspondientes imágenes del espectro infrarrojo cercano NIR, adicionalmente se generaron para las imágenes visibles e infrarrojas sus respectivas imágenes siluetas de los defectos. Al final de este módulo se aplica data augmentation para incrementar el número total final de imágenes del dataset.

El módulo 2 consistió en el entrenamiento de los modelos seleccionados de redes neuronales convolucionales (CNN) utilizando el dataset previamente construido en el módulo 1. Para el entrenamiento se utilizaron dos esquemas: 1) entrenamiento mediante propagación directa (Feedforward) para las imágenes RGB e infrarrojo cercano (NIR), y 2) un esquema siamés para entrenar con las imágenes RGB y NIR junto con sus respectivas siluetas. Los

modelos de CNN entrenados fueron: ResNet50, MobileNetV1, DenseNet121 y VGG19. Finalmente, el tercer módulo se centra en la etapa de evaluación de los resultados obtenidos con cada modelo de CNN, donde los CNN entrenados son validados cuantitativamente utilizando métricas de precisión para medir su capacidad de detección de los defectos en la fruta de estudio (manzana).



**Figura 2.1 Módulos de desarrollo de la solución**

## 2.2 Módulo 1: Creación del dataset

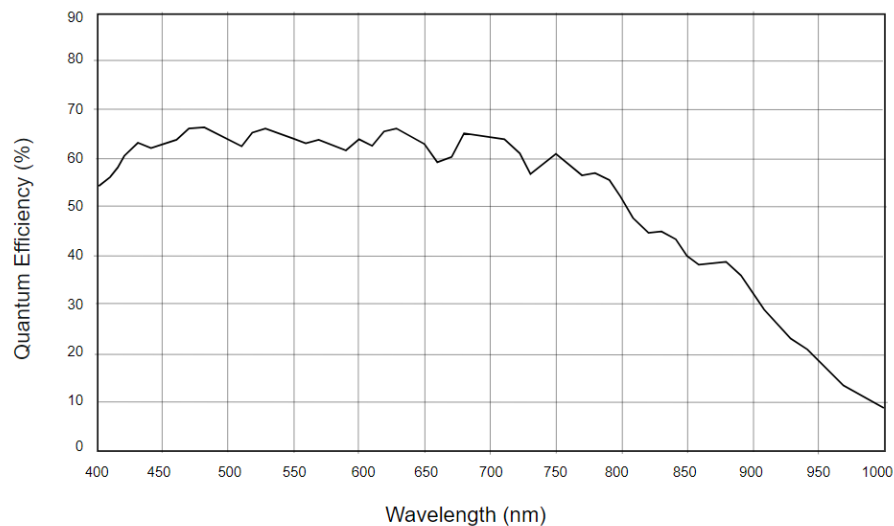
### 2.2.1 Adquisición de imágenes

Para la captura de imágenes se emplearon los equipos otorgados por el Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS), lo cual estuvo conformado por dos cámaras: una cámara RGB para adquirir imágenes en el espectro visible y otra cámara NIR para obtener imágenes en el espectro infrarrojo cercano. La iluminación de la zona de captura se realizó mediante lámparas LED que proveían una iluminación difusa de 1170

lúmenes, logrando una iluminación uniforme de la superficie de la fruta, en nuestro caso las manzanas.

La cámara empleada para la captura de imágenes en el espectro visible (RGB) fue una cámara Basler GigE con una resolución de 1280x960 y 8 bits/píxel con un *frame rate* de 60fps y ganancia de brillo estática configurada en 18 dB [10]. La adquisición de las imágenes se automatizó mediante un script de Python que permitía el disparo programado de la cámara, seguido del almacenamiento de las imágenes en sus respectivos directorios, tanto para las imágenes RGB como para las imágenes NIR.

La captura de las imágenes en el espectro infrarrojo cercano (NIR) se realizó con una cámara Basler ace acA1300-60gmNIR con una resolución de 1280x1024 píxeles y 8 bits/píxel con un frame rate de 60fps y ganancia de brillo estática configurada en 18 dB. La curva de respuesta espectral de esta cámara se muestra en la Figura 2.2, lo que demuestra una eficiencia cuántica menor al 70% y mayor al 40% para el rango de 400 – 800 nm [10].



**Figura 2.2 Curva de respuesta espectral de cámara NIR**

Dado que la cámara NIR utilizada para la captura de imágenes en el espectro infrarrojo cercano (NIR) tiene una respuesta espectral que abarca desde los 400 nm hasta los 1000 nm, donde gran parte de este rango está en el espectro visible (400 a 700 nm), se decidió emplear un filtro de paso de banda disponible para limitar la captura de luz en el rango de 640 nm a 680 nm. Aunque este filtro no es el óptimo para el espectro infrarrojo cercano, fue la opción disponible en ese momento. El filtro de paso de banda permitió reducir la influencia de la luz visible no deseada y concentrarse en la captura de la radiación cercana al infrarrojo NIR. La Figura 2.3 muestra un par de imágenes RGB y NIR correspondientes a la captura de una escena de la fruta (manzana).



**Figura 2.3 Imagen RGB y NIR con defecto golpe**

### **2.2.2 Etiquetado de dataset**

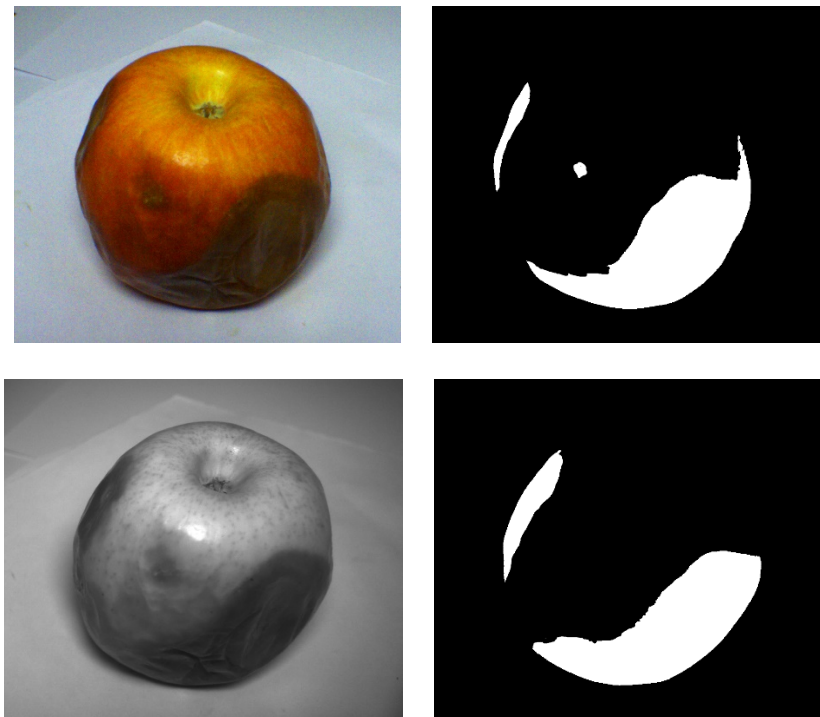
Para facilitar el posterior procesamiento de las imágenes, se implementó un sistema de etiquetado y almacenamiento normalizado. Las imágenes capturadas en el espectro visible (RGB) se guardaron en formato PNG de tres bandas (rojo, verde y azul), mientras que las imágenes del espectro infrarrojo cercano (NIR) se almacenaron en una única banda.

El nombre de cada archivo PNG tiene el formato de espectro electromagnético de adquisición (visible o NIR), seguido de un guion bajo y número consecutivo de la imagen. Por ejemplo, para la tercera imagen de una manzana con manchas capturada en el espectro visible, el

nombre sería: rgb\_0003.png y estaría localizada en el directorio de “apple/spots”, correspondiente al directorio de manzanas con manchas.

### 2.2.3 Preprocesamiento de imágenes

Una de las etapas clave de este proceso es la segmentación, donde se lleva a cabo la subdivisión de la imagen en regiones o segmentos significativos. En el contexto específico de este proyecto, donde las imágenes fueron capturadas sobre un fondo blanco con un sistema de iluminación difusa, la segmentación se enfocó en la identificación clara y precisa de la silueta de los defectos. La Figura 2.4 muestra un ejemplo de la silueta de los defectos detectados tanto para una imagen RGB como NIR.



**Figura 2.4** Imágenes RGB y NIR con su respectiva silueta

Al tener control sobre la adquisición se permite minimizar sombras y tratar de mejorar la visibilidad de los defectos evitando someter a las imágenes a procesamiento adicional para ajustar el contraste, corregir posibles variaciones de iluminación, etc.

Par generar las siluetas de los defectos tanto para las imágenes RGB como para las imágenes NIR se optó por emplear el modelo *Segmentation of Anything* (SAM), una herramienta avanzada que combina técnicas de umbralización y segmentación, basadas en color y características texturales [11]. Dada la captura de imágenes sobre un fondo blanco, SAM aprovechó la diferencia de color entre la fruta y el fondo, aplicando técnicas de umbralización para separar de manera clara la fruta del fondo.

SAM utiliza algoritmos específicos, incluyendo el método de crecimiento de regiones y enfoques basados en aprendizaje automático, para lograr una segmentación precisa, tomando en consideración la complejidad de los defectos y la variabilidad de las frutas capturadas [11] [12].

Se ajustaron los parámetros de SAM según fuera necesario para optimizar la precisión y minimizar posibles falsos positivos o falsos negativos.

Para casos especiales donde SAM no segmente adecuadamente la región deseada, se usará herramientas que provee el mismo SAM para segmentar áreas específicas de la imagen.

#### **2.2.4 Data augmentation**

La data augmentation (aumento de datos) representó una técnica fundamental en el aprendizaje profundo para conferir robustez a los modelos y evitar el sobreajuste (overfitting).

Esta técnica se empleó para generar, de manera aleatoria, variaciones sintéticas en los datos de entrenamiento existentes.

Esta estrategia se consideró especialmente recomendable en casos de conjuntos de entrenamiento pequeños o desequilibrados. Al aplicar transformaciones, como rotaciones, giros, recortes y ajustes de brillo/contraste en las imágenes, se logró efectivamente multiplicar el tamaño del conjunto de datos de imágenes.

En este proyecto, el conjunto original de imágenes con los 3 tipos de defectos era limitado en relación con su tamaño, esto es, inicialmente cada defecto estaba compuesto por 3000 imágenes RGB, 3000 NIR, y sus correspondiente imágenes siluetas de defecto, totalizando 12.000 imágenes para cada clase de defectos. La aplicación de data augmentation generó variaciones artificiales, incrementando el número a 5000 imágenes por cada clase y, por ende, alcanzando un total de 60 000 imágenes.

Esta medida permitió el entrenamiento de modelos más efectivos al disponer de una mayor diversidad de datos, mitigando el problema del overfitting. La data augmentation mejoró el rendimiento general al exponer el modelo a una gama más amplia de escenarios durante el entrenamiento, destacándose como una técnica esencial en proyectos con recursos de datos limitados.

### **2.3 Módulo 2: Entrenamiento de modelos**

En este módulo, se describió la metodología de entrenamiento para los modelos de CNN usados en este trabajo. Para ello, se dispuso del equipo de cómputo proporcionado por el CIDIS, que permitió el desarrollo de los entrenamientos masivos requeridos. La elección de Python se



fundamenta en las implementaciones optimizadas ofrecidas por destacadas librerías, entre las cuales se resalta TensorFlow.

Los modelos ResNet50, DenseNet121, MobileNetV1 y VGG19, seleccionados antes para esta tarea, se entrenaron de forma independiente para clasificar los 3 tipos de defectos de la manzana. La partición de datos se realizó con un esquema 95/5, donde el 95% de los datos se destinó al entrenamiento, el 5% a la validación durante el entrenamiento. La división garantizó tener datos adecuados para el entrenamiento y la validación de cada modelo considerado.

Se utilizó un ordenador con una tarjeta gráfica RTX 3060, 32GB de memoria RAM

Para el entrenamiento de estos modelos se usaron dos enfoques de arquitecturas, esto es: feedforward y siamesa, los cuales se describen a continuación.

### **2.3.1 Arquitectura de red tipo Feedforward**

En la revisión literaria, se determinó que un enfoque con redes neuronales Feedforward resultó adecuado para la clasificación de imágenes. Estudios previos [10][11] demostraron el éxito en la utilización de redes convolucionales Feedforward para identificar diferentes defectos visuales.

Este enfoque fue recomendado para tareas de clasificación debido a su capacidad de aprender patrones secuenciales en las imágenes. No se requería información previa ni comparaciones entre ejemplos.

En el marco de este proyecto, se implementó una arquitectura convolucional Feedforward para analizar por separado las imágenes RGB y NIR. Esto permitió explotar mejor las características visuales capturadas en cada espectro, tal como se muestra en la Figura 2.5.

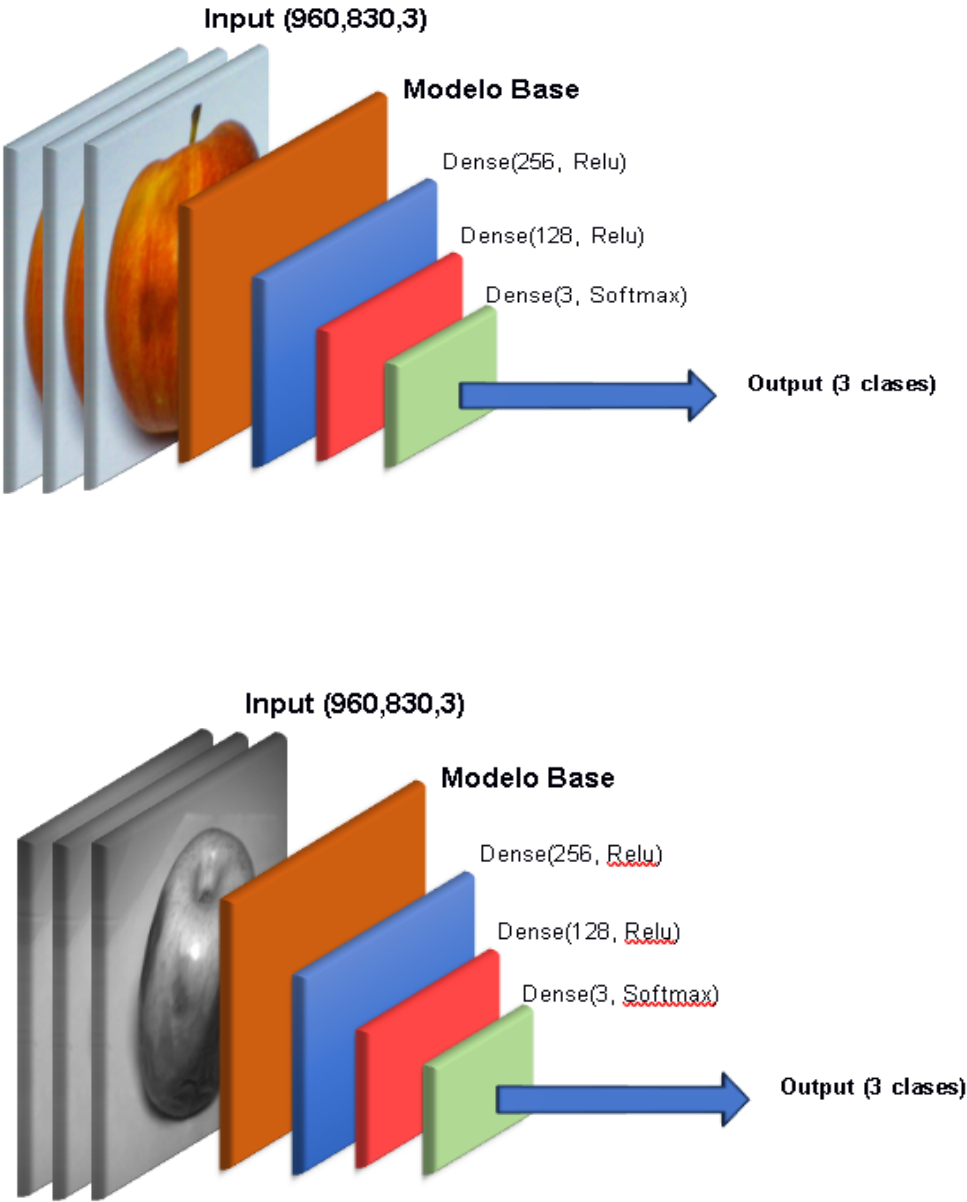


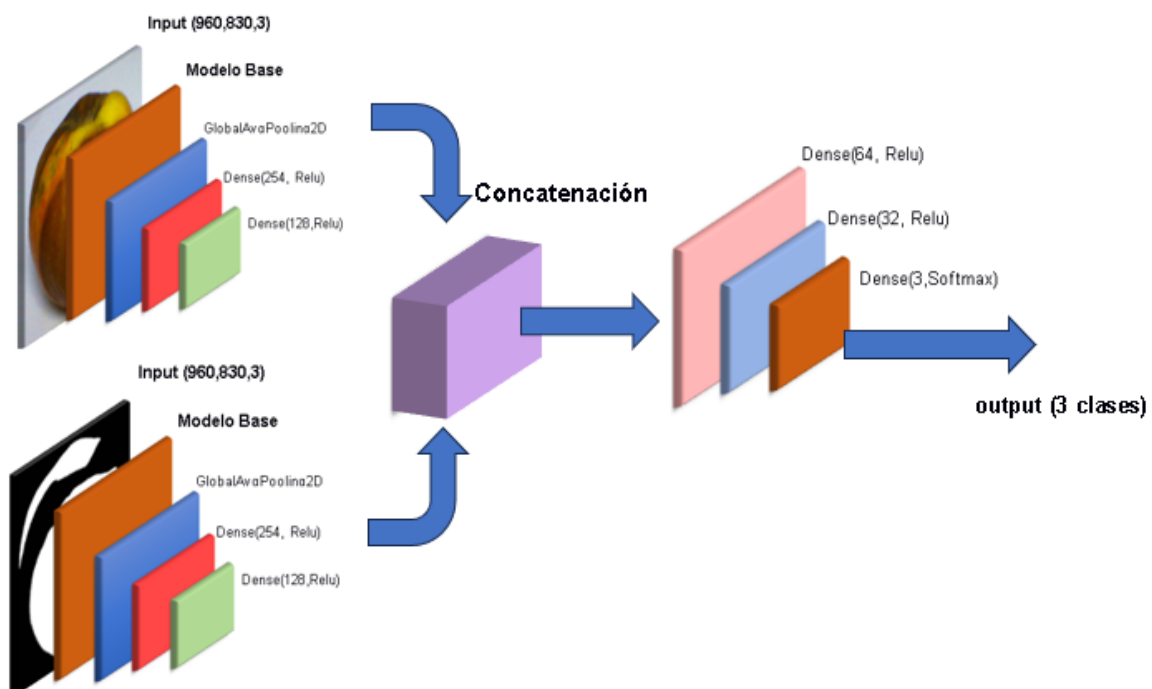
Figura 2.5 Arquitecturas Feedforwards propuestas

En detalle, se entrenaron dos modelos idénticos, uno para RGB y otro para NIR. Cada modelo clasificó los defectos presentes en las imágenes de su dominio espectral correspondiente (visible o infrarrojo cercano).

Este enfoque particionado aprovechó mejor las ventajas del Feedforward y las diferencias entre los espectros, optimizando el aprendizaje para cada modalidad. La clasificación individual redundó en una detección más efectiva.

### 2.3.2 Arquitectura de red siamesa

La investigación llevó a considerar el uso de una arquitectura Siamesa para comparar y analizar similitudes y diferencias entre pares de imágenes. Este tipo de arquitectura recibió como entrada un par de imágenes y extrajo características que comparó para determinar su similitud. Se usaron 3 posibles combinaciones de pares de imágenes como entrada para la arquitectura Siamesa, estos pares fueron: 1) RGB + silueta defecto/RGB; 2) NIR + silueta defecto/NIR; y 3) RGB + NIR; tal como se detalla en la Figura 2.6. En este caso, se ingresaron a cada rama imágenes de frutas con y sin defectos segmentados respectivamente. La red aprendió a identificar patrones de discrepancia que indicaban la presencia de imperfecciones.



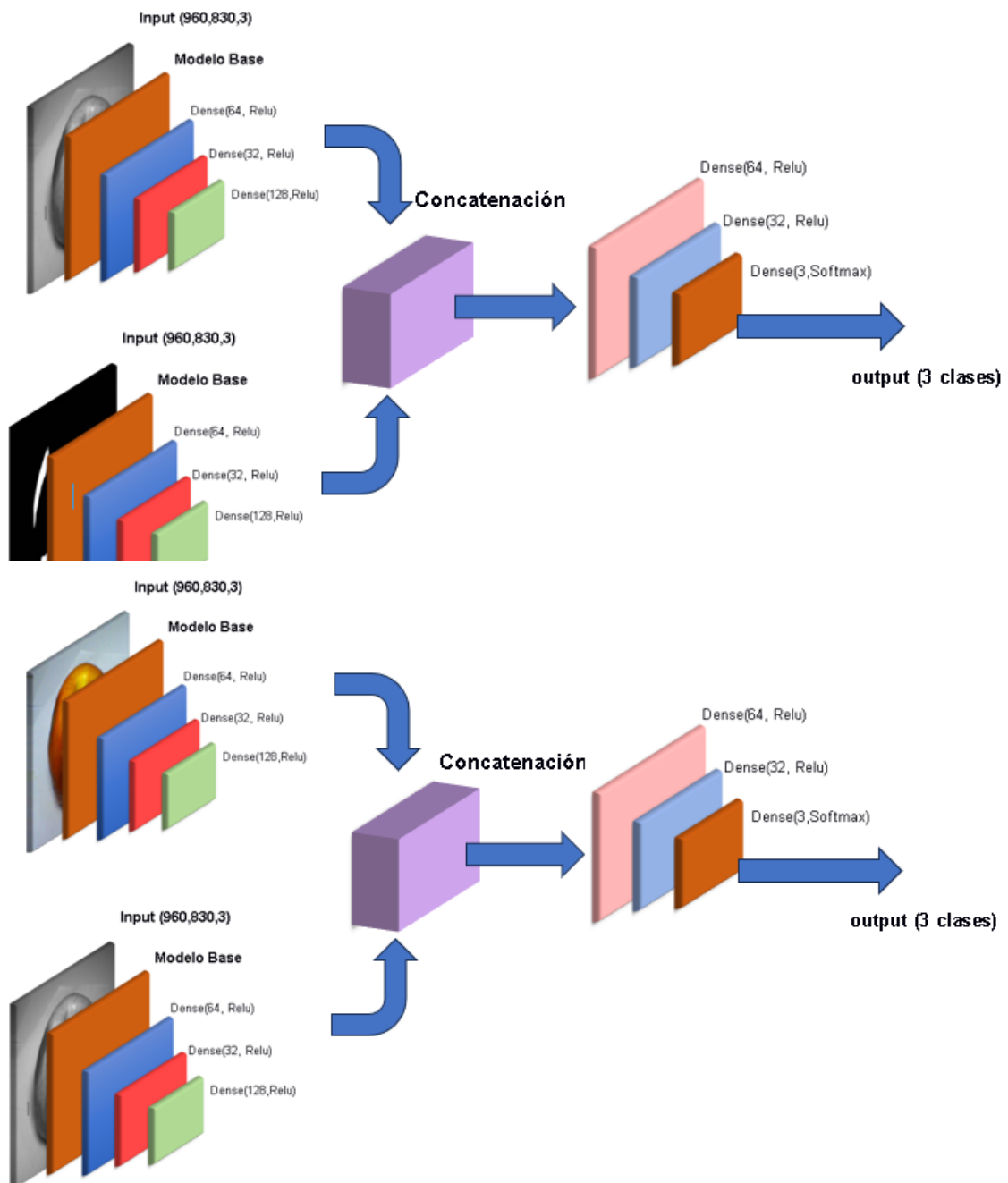


Figura 2.6 Arquitecturas Siamesas propuestas

Una vez extraídas las características por cada rama, una capa de fusión basada en Perceptrones Multicapa (MLP por sus siglas en inglés) unificó la información en una representación combinada de alto nivel semántico. La MLP permitió modelar complejas relaciones no lineales entre las características de ambas imágenes, lo cual fue crucial para la detección precisa de defectos.

## **2.4 Módulo 3: Evaluación de modelos**

Este módulo se centra en la evaluación de los modelos, destacando diversas métricas utilizadas para la evaluación de los modelos seleccionados, tales como se detallan abajo.

### **Matriz de confusión:**

La matriz de confusión nos mostrará detalladamente los aciertos y errores del clasificador en una tabla de contingencia. Nos indicará cuántas muestras fueron correctamente predichas como defectuosas/no defectuosas, y cuántas fueron incorrectamente clasificadas.

### **Accuracy (precisión):**

El *accuracy* revela el porcentaje de muestras correctamente clasificadas sobre el total.

### **Precision:**

La *precision* evalúa la probabilidad de que una muestra predicha como ‘defectuosa’, lo sea realmente, calculada como los verdaderos positivos sobre la suma de verdaderos y falsos positivos.

**Recall (sensibilidad):**

El Recall indica el porcentaje de muestras defectuosas reales detectadas, calculado como los verdaderos positivos sobre la suma de verdaderos positivos y falsos negativos.

## **Capítulo 3**

### **3. ANÁLISIS Y RESULTADOS**

En este capítulo se detallarán los resultados obtenidos en cada módulo del diseño de la solución. Se describirá la obtención del dataset de imágenes RGB y NIR. Además, se explicará el proceso de segmentación utilizado para obtener las siluetas de los defectos correspondientes en las imágenes.

Se expondrán los resultados obtenidos en el entrenamiento de los modelos de redes neuronales convolucionales (CNN) seleccionados. Se proporcionarán detalles sobre el enfoque de entrenamiento empleado, incluyendo el enfoque Feedforward y el enfoque de red Siamesa.

Finalmente, se evaluarán los modelos entrenados. Se discutirán los experimentos explorados y se presentarán los resultados obtenidos en términos de métricas de desempeño, como accuracy y recall.

#### **3.1 Creación del conjunto de imágenes**

##### **3.1.1 Dataset de imágenes RGB y NIR**

Para la obtención de las imágenes RGB y NIR, se utilizó el equipo y las cámaras proporcionadas por el Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS). Se seleccionaron manzanas que representaban las tres clases de defectos a clasificar, esto es: golpes, manchas y podredumbre. La configuración y colocación de las cámaras permitió agilizar la adquisición, obteniendo 160 imágenes en RGB y NIR para cada fruta.



Para la obtención de imágenes con defecto de golpes (bruise), se utilizaron 6 manzanas que fueron sometidas a golpes leves. Se registró el avance de los golpes durante los siguientes 6 días después de haber provocado los impactos. Es importante destacar que, en el primer día, los golpes no eran visibles en ningún espectro, mientras que en los siguientes días los golpes o magulladuras en la fruta se iban haciendo más visibles cada día.

Este proceso de seguimiento y registro permitió obtener un conjunto de imágenes RGB y NIR significativo. En total, se obtuvieron alrededor de 940 imágenes por día en cada espectro (RGB y NIR) a lo largo de los 6 días de seguimiento para las 6 manzanas utilizadas, descartando las imágenes del primer día debido a que el defecto en el primer día no era perceptible bajo ningún espectro. Después, se realizó un proceso de selección de imágenes válidas para el entrenamiento, lo que resultó en imágenes para el espectro RGB y para el espectro NIR, como se muestra en la Tabla 3.1.

	<b>Golpes</b>	
	<b>RGB</b>	<b>NIR</b>
<b>día 1</b>	940	940
<b>día 2</b>	923	923
<b>día 3</b>	917	917
<b>día 4</b>	869	869
<b>día 5</b>	905	905
<b>día 6</b>	925	925
<b>Total</b>	<b>4539</b>	<b>4539</b>

**Tabla 3.1 Cantidad de imágenes RGB y NIR con defecto de golpes.**

Para la obtención de las imágenes con el defecto de manchas (*spots*), se utilizaron 20 frutas con este defecto. Estas manchas consistían en puntos blancos visibles a simple vista. Al capturar imágenes, se observó una variación en el espectro NIR, donde las manchas se visualizaban como puntos negros, a diferencia de los puntos blancos visibles en el espectro RGB.

En la obtención de imágenes se consideró esta diferencia en la apariencia de las manchas entre los dos espectros.

Tras adquirir las imágenes, se hizo una selección de las adecuadas para entrenar a los modelos. Este proceso de selección permitió obtener un conjunto final de 3,194 imágenes tanto en el espectro RGB como en el NIR para el defecto de manchas, tabla 3.2.

Para el defecto de podredumbre (*rot*), se utilizaron 20 frutas que fueron sometidas a un proceso controlado de pudrición. Este proceso permitió desarrollar los defectos de manera que fueran lo más visibles posible en las frutas. Tras seleccionar las imágenes adecuadas, se obtuvieron 3,094 imágenes en el espectro RGB y en el NIR para el defecto de podredumbre como se muestra en la Tabla 3.2.

<b>Manchas</b>		<b>Pudrición</b>	
<b>RGB</b>	<b>NIR</b>	<b>RGB</b>	<b>NIR</b>
<b>3136</b>	3136	3094	3094

**Tabla 3.2 Cantidad de imágenes RGB y NIR con defectos de manchas y pudrición.**

Es importante mencionar que inicialmente se intentó incluir una cuarta clase de manzanas sin defectos en el estudio. La adquisición de frutas sin defectos resultaba complicada porque los puntos (defecto de manchas) eran una característica constante en todas las manzanas, independientemente de su variación. Esta situación refleja la realidad del mercado ecuatoriano, donde las manzanas suelen presentar puntos como parte de su apariencia normal.

Esta dificultad para obtener manzanas sin defectos reforzó la decisión de centrarse en las tres clases de defectos mencionadas anteriormente: golpes, manchas y podredumbre. Estas son las principales problemáticas encontradas en las manzanas del mercado ecuatoriano y, por lo tanto, resultaban fundamentales de abordar en el estudio.

### **3.1.2 Dataset de imágenes siluetas en el espectro RGB y NIR**

Para la obtención de las siluetas de los defectos, se utilizó la herramienta *Segmentation Anything Model (SAM)*. Sin embargo, debido a las particularidades de cada caso, se aplicaron enfoques específicos.

SAM es una herramienta que funciona mejor con imágenes que presentan variaciones bruscas en el color o la intensidad del brillo. Sin embargo, para el defecto de golpes, que es más sutil en comparación con otros defectos, fue un desafío lograr que SAM detectara y segmentara correctamente el defecto. Aunque el golpe era visible en la imagen, SAM no lograba identificarlo adecuadamente, lo que dificultaba la obtención de la silueta deseada.

Para abordar esta situación, se implementó una variación de la herramienta *SAM* utilizando *SAM Predictor*. Se desarrolló un script que permitía seleccionar manualmente un área específica en cada imagen, tanto en el espectro RGB como en el NIR. Luego, se utilizó *SAM Predictor* con los archivos de las áreas previamente seleccionadas para obtener automáticamente las siluetas correspondientes en cada imagen RGB y NIR.

Este enfoque permitió superar las limitaciones de SAM en la detección de los golpes y obtener las siluetas deseadas de forma más precisa. Al combinar la selección manual de áreas con la herramienta *SAM Predictor*, se logró obtener siluetas más representativas de los defectos en las imágenes tanto en el espectro RGB como en el NIR.

Para obtener las siluetas de los defectos de manchas, se modificaron los parámetros de SAM para generar máscaras con segmentaciones muy pequeñas que se correspondieran con las siluetas de estos defectos. Esta adaptación de los parámetros permitió obtener segmentaciones más precisas y detalladas de las manchas en las imágenes.

Sin embargo, es importante tener en cuenta que este proceso de segmentación con parámetros modificados requirió un alto consumo de recursos computacionales. Por la complejidad de la tarea y la necesidad de procesar muchas imágenes, el proceso de segmentación para este defecto tomó 6 días, durante los cuales el equipo de cómputo trabajó de manera continua.

A pesar del tiempo y los recursos requeridos, al final del proceso se lograron obtener las siluetas respectivas tanto para las imágenes RGB como para el NIR. Estas siluetas representan de manera precisa las áreas afectadas por las manchas en cada imagen.

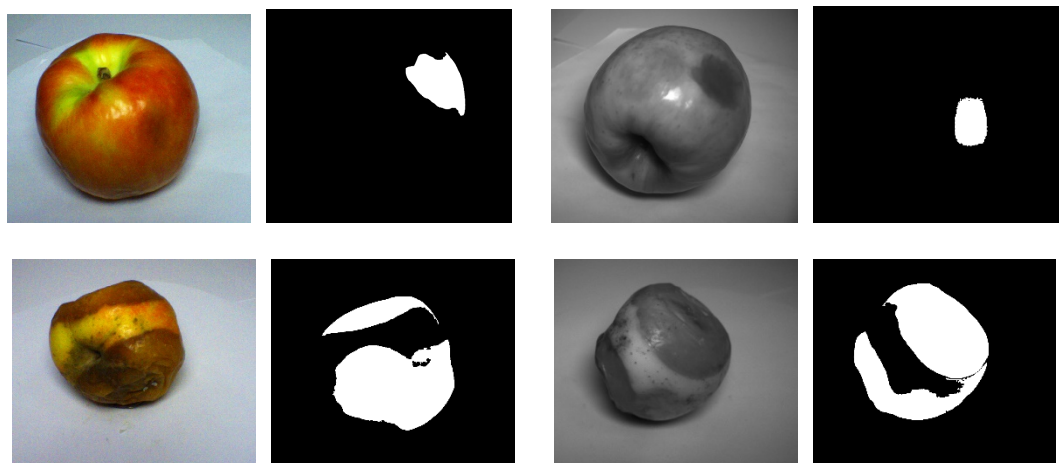
Para los defectos de podredumbre, se utilizó una variación del código empleado para las manchas, modificando los parámetros de SAM. El objetivo fue evitar que SAM generara máscaras con segmentaciones pequeñas y filtrando las áreas de las segmentaciones para obtener las siluetas correspondientes a los defectos de podredumbre.

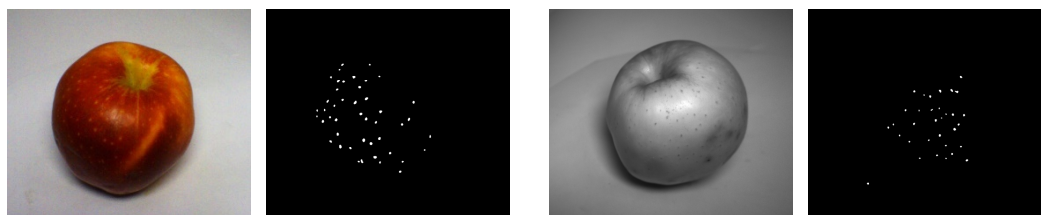
Los defectos de podredumbre presentaban casos especiales en los que se necesitaban configuraciones particulares por las dimensiones del defecto. En ocasiones, el área afectada por

la podredumbre era más grande que la parte sana de la fruta. Para abordar estos casos, se adaptaron las configuraciones y parámetros del proceso de segmentación de manera específica.

Al final del proceso, se logró obtener las siluetas respectivas a cada espectro, tanto en el RGB como en el NIR, para los defectos de podredumbre. Estas siluetas permiten delimitar de forma precisa las áreas afectadas por la podredumbre en las imágenes.

Esta adaptación del código y el ajuste de parámetros, junto con la consideración de los casos especiales, fueron fundamentales para obtener siluetas adecuadas de los defectos de podredumbre. Cada defecto presenta sus propias características y desafíos, por lo que es necesario realizar ajustes y adaptaciones específicas para obtener resultados óptimos en el proceso de segmentación. En la siguiente Figura 3.1 se muestra un ejemplo de la silueta generada para cada uno de los 3 tipos de defectos estudiados en este proyecto, esto es: golpes, manchas y podredumbre.





**Figura 3.1 Silueta resultantes generadas para las imágenes RGB y NIR**

### 3.1.3 Data augmentation

Debido a la complejidad en la adquisición de imágenes, se decidió aplicar técnicas de aumento de datos (*data augmentation*) para aumentar la cantidad de imágenes disponibles para cada defecto tanto en el espectro RGB como en el NIR. El objetivo era alcanzar 50.000 imágenes por defecto.

Para lograr esto, se desarrolló un script en Python que automatizó el proceso de *data augmentation*. En particular, se aplicó una operación de *flip* horizontal a las imágenes capturadas para generar nuevas imágenes. Esta operación de *flip* horizontal implica reflejar la imagen horizontalmente, lo que crea una nueva imagen con una apariencia similar, pero con una orientación opuesta.

El script seleccionó de forma aleatoria las imágenes existentes y aplicó el *flip* horizontal a la cantidad necesaria para completar la cantidad de imágenes objetivo. De esta manera, se generaron nuevas imágenes que enriquecieron el conjunto de datos disponible.

Es importante destacar que este proceso de *data augmentation* también se realizó en las siluetas correspondientes a cada imagen. Esto garantizó que hubiera una correspondencia

adecuada entre las imágenes y las siluetas de los defectos, lo que resulta esencial para el análisis y la clasificación precisa.

## **3.2 Arquitectura de los modelos**

### **3.2.1 Justificación de selección de modelos**

Los modelos ResNet50, DenseNet121, MobileNetV1 y VGG19 se eligieron en base a los resultados reportados en el estudio [13], ya que este trabajó específicamente detectando defectos (magulladuras) en manzanas utilizando imágenes en el espectro del infrarrojo cercano (NIR), que también se usó en este estudio.

En el estudio mencionado se demostró la viabilidad de detectar magulladuras en manzanas utilizando imágenes en el espectro del infrarrojo cercano (NIR). Específicamente, se identificaron las siguientes longitudes de onda como óptimas para distinguir tejidos sanos de lesionados: 750nm, 850nm y 960nm.

Asimismo, evaluó el desempeño de varios modelos al realizar esta tarea de detección. Entre ellos se encontraba el modelo Densenet-121 [23], el cual obtuvo un alto promedio de 97,8% en el coeficiente de Matthews al validar de manera cruzada. Este resultado indica que casi todas las manzanas fueron correctamente clasificadas.

Otro modelo evaluado fue ResNet50 [22] que también reportó un alto rendimiento al alcanzar 97% de precisión. Los resultados demostraron la capacidad de ambos modelos, particularmente Densenet-121 y ResNet50, para distinguir con precisión manzanas sanas de las dañadas trabajando con imágenes NIR.

Dado que dicho estudio evaluó específicamente la detección de defectos en manzanas usando imágenes NIR, al igual que en el presente estudio, se seleccionaron los modelos DenseNet121, ResNet50 y VGG19 por haber demostrado previamente detectar con éxito este tipo de defectos con este tipo de imágenes

### **3.2.2 Hiperparámetros utilizados para el entrenamiento**

Para todos los modelos se utilizó la arquitectura de su respectivo modelo preentrenado y se realizaron ajustes de capas superiores para la tarea específica de clasificación de defectos en manzanas. Las modificaciones en los hiperparámetros fueron las siguientes:

Se incorporaron dos capas completamente conectadas con 256 y 128 neuronas, respectivamente, en la arquitectura del modelo. Estas capas, que forman parte de la red personalizada construida sobre la base de cada modelo, se agregaron a continuación de la capa de pooling global proveniente de la arquitectura MobileNet.

Para la activación de las capas ocultas, se optó por ReLU (Rectified Linear Unit), mientras que Softmax se utilizó como función de activación en la capa de salida para la clasificación multiclase [19].

El optimizador seleccionado fue Adam, configurado con una tasa de aprendizaje inicial de 0.001. Se implementó una reducción de la tasa de aprendizaje (decay) del 5% (0.95) después de cada época para mejorar la convergencia del modelo. Además, se aplicó un Dropout del 50% en las capas completamente conectadas para mitigar posibles problemas de sobreajuste.



Se utilizó la función *flow\_from\_directory* desde un directorio base para la distribución de clases. Sus parámetros fueron “*batch\_size = 8*”, “*class\_mode = categorical*” para que retorne etiquetas en formato 2D *one-hot encoded* y “*target\_size = (960,30)*” por el tamaño de entrada de las imágenes del *dataset* [20].

El tamaño del lote utilizado durante el entrenamiento se limitó a 8 imágenes por restricciones de rendimiento, relacionadas con la capacidad de la tarjeta gráfica utilizada. La imposibilidad de aumentar el tamaño del lote a 32 o 16 se debió a un error de falta de memoria al procesar imágenes de dimensiones 960x830. Estas dimensiones, al ser relativamente grandes, generaron una carga significativa en la memoria de la GPU, lo que resultó en la necesidad de reducir el tamaño del lote para evitar el agotamiento de recursos y fallos durante el entrenamiento del modelo.

La función de pérdida fue “*categorical\_crossentropy*”, la cual es adecuada para problemas de clasificación multiclase. En el generador de datos se utilizó un “*rescale*” de 1./255, para dividir los valores de cada píxel por 255. Esto se realizó para escalar los valores de los píxeles de las imágenes y normalizarlos en el rango [0, 1]. Además, esto permite que el modelo converja más rápido y de manera más efectiva, ya que las actualizaciones de peso serán más estables y eficientes [20].

Se utilizaron 25 épocas para el entrenamiento de cada modelo. En casos particulares en los que la curva del *accuracy* seguía aumentando, como en MobileNetV1, se cargaron los pesos de la mejor época y se disminuyó el *learning rate* a 0.00005 para mejorar la predicción del modelo.

### **3.3 Entrenamiento de modelos**

#### **3.3.1 Proceso de entrenamiento de modelos *feedforward***

El proceso de entrenamiento para todos los modelos fue el de la selección y preparación del conjunto de datos, definición de la arquitectura del modelo (MobileNetV1, Densenet121, Resnet50 o VGG19), definición de la cantidad de neuronas en cada capa y las funciones de activación, definición de la función de pérdida y el optimizador a utilizar.

Se configuró un generador de datos para cargar y preprocesar las imágenes antes de alimentarlas al modelo para hacer un “rescale” de las imágenes del dataset. Luego se compiló el modelo con el optimizador, tipo de pérdida definido y la métrica de accuracy para la evaluación de desempeño del modelo.

Se aprovechó el uso de la técnica de transfer learning para mitigar los efectos de lotes pequeños, aprovechando el conocimiento previo de los modelos preentrenados en los conjuntos disponibles.

Se utilizó el conjunto de entrenamiento para entrenar el modelo. Inicialmente se utilizaron 10 épocas para estimar el tiempo de entrenamiento y las métricas obtenidas para validar que no haya problemas en la carga de imágenes por clase. Luego de esta validación se utilizó 15 épocas y dado que la curva de pérdida en el conjunto de entrenamiento disminuía y no alcanzaba una meseta, se aumentó el número de épocas a 25. Ya con 25 épocas la curva de precisión en el conjunto de entrenamiento se mantenía estable. Para valores mayores provocaba sobreajuste y la curva de precisión divergía para una cantidad muy alta de épocas [20].

Se utilizó un conjunto de validación para evaluar el rendimiento del modelo durante el entrenamiento. Se utilizaron 500 imágenes para validación por clase, con un total de 1500 para las 3 clases definidas. Esto con el fin de detectar el sobreajuste y ajustar los hiperparámetros en base a los resultados de la validación.

Luego se realizaron evaluaciones con el conjunto de prueba una vez se completó el entrenamiento, para evaluar el rendimiento final del modelo en datos no vistos.

Para las arquitecturas Resnet50 y Densenet121, se realizó un postentrenamiento tomando los pesos de la época 25 de su respectivo modelo entrenado, para mejorar el *accuracy* obtenido en el conjunto de validación, el cual fue 77,74% para Densenet121 y 82,22% para Resnet50. El postentrenamiento implicó un entrenamiento de 10 épocas y una disminución del *learning rate* en 0.0005, lo que mejoró significativamente el *accuracy* del conjunto de validación sin afectar sustancialmente el *accuracy* del conjunto de entrenamiento.

### **3.3.2 Proceso de entrenamiento de modelos siamesa**

El proceso de entrenamiento para modelos en arquitectura siamesa utilizando las redes VGG19, MobileNetV1, DenseNet121 y ResNet50 se llevó a cabo con configuraciones consistentes para poder comparar los resultados con los modelos previamente entrenados en una arquitectura Feedforward.

Se emplearon conjuntos de imágenes específicos para cada tipo de par RGB-silueta (RGB, NIR) y la combinación de RGB y NIR. Cada conjunto incluyó 4,000 imágenes por cada clase (golpes, manchas, podredumbre) destinadas al entrenamiento, con 500 para pruebas y 500

para validación. Esto resultó en un total de 12,000 imágenes para cada rama en el entrenamiento de cada par, garantizando una representación equilibrada de las clases.

Se adoptó una configuración uniforme para el entrenamiento, basada en valores previamente establecidos durante el entrenamiento de modelos Feedforward. Se utilizó un tamaño de lote (batch size) de 8 y se realizaron 25 épocas para mantener la coherencia en la comparación de resultados.

Se implementó un generador de pares de imágenes que, en cada iteración, seleccionaba dos imágenes de diferentes modalidades (RGB, NIR) junto con su respectiva silueta, creando los pares RGB-silueta, NIR-silueta y RGB-NIR.

Con los modelos base seleccionados, se construyó un modelo siamés para cada arquitectura base, con dos ramas compartidas que procesaban las imágenes por separado. Cada rama consistía en la arquitectura base más capas adicionales.

En la rama compartida de cada modelo, se agregaron capas de Global Average Pooling, Batch Normalization y capas densas con ReLU para extraer características antes de la concatenación.

Se configuraron los modelos para extraer características de las imágenes RGB y NIR antes de su concatenación, utilizando esta salida como entrada a capas densas MLP para la clasificación final.

Cada modelo siamesa se compiló con Adam, `categorical_crossentropy` y `accuracy`, entrenándolos con el generador de pares.

Finalmente, se evaluaron los modelos en el conjunto de prueba para determinar su precisión, permitiendo la comparación de resultados entre las diferentes arquitecturas.

### **3.3.3 Evaluación de métricas de rendimiento durante el entrenamiento y validación**

Durante el desarrollo de la solución, se llevaron a cabo exhaustivas evaluaciones de rendimiento para cada uno de los modelos entrenados (Resnet50, Densenet121, MobilenetV1 y VGG19) en ambas arquitecturas (Feedforward, Siamesa), tanto en el conjunto de entrenamiento como en el conjunto de validación. Se utilizaron métricas estándar para evaluar la eficacia y la capacidad de generalización de los modelos.

#### **Métricas de Entrenamiento**

##### **Accuracy (Precisión)**

El accuracy se calculó como el porcentaje de predicciones correctas respecto al total de muestras en el conjunto de entrenamiento. Este indicador da una visión del rendimiento del modelo durante el entrenamiento.

##### **Pérdida (Loss)**

La función de pérdida se utilizó como medida de cuán bien el modelo estaba aprendiendo durante el entrenamiento. Se monitoreó el descenso de la pérdida a lo largo de las épocas para asegurar un aprendizaje adecuado.

##### **Curvas de Aprendizaje (Learning Curves)**

Las curvas de aprendizaje se generaron para visualizar la evolución de la precisión y la pérdida durante el entrenamiento. Esto ayudó a identificar posibles problemas de overfitting o underfitting (subajuste). Se usó para verificar la eficiencia del aumento o disminución de las épocas de entrenamiento utilizadas. Además, se utilizó para la decisión de postentrenamiento de los modelos feedforward Resnet50 y Densenet121.

### **Métricas de Validación:**

#### **Precisión en Validación (Validation Accuracy)**

La precisión en el conjunto de validación proporciona una evaluación del rendimiento del modelo en datos no vistos. Se comparó con la precisión en el conjunto de entrenamiento para identificar posibles problemas de sobreajuste.

#### **Pérdida en Validación (Validation Loss)**

La función de pérdida en el conjunto de validación se utilizó para evaluar la capacidad de generalización del modelo. Un aumento en la pérdida de validación podría indicar overfitting.

#### **Matriz de Confusión en Validación**

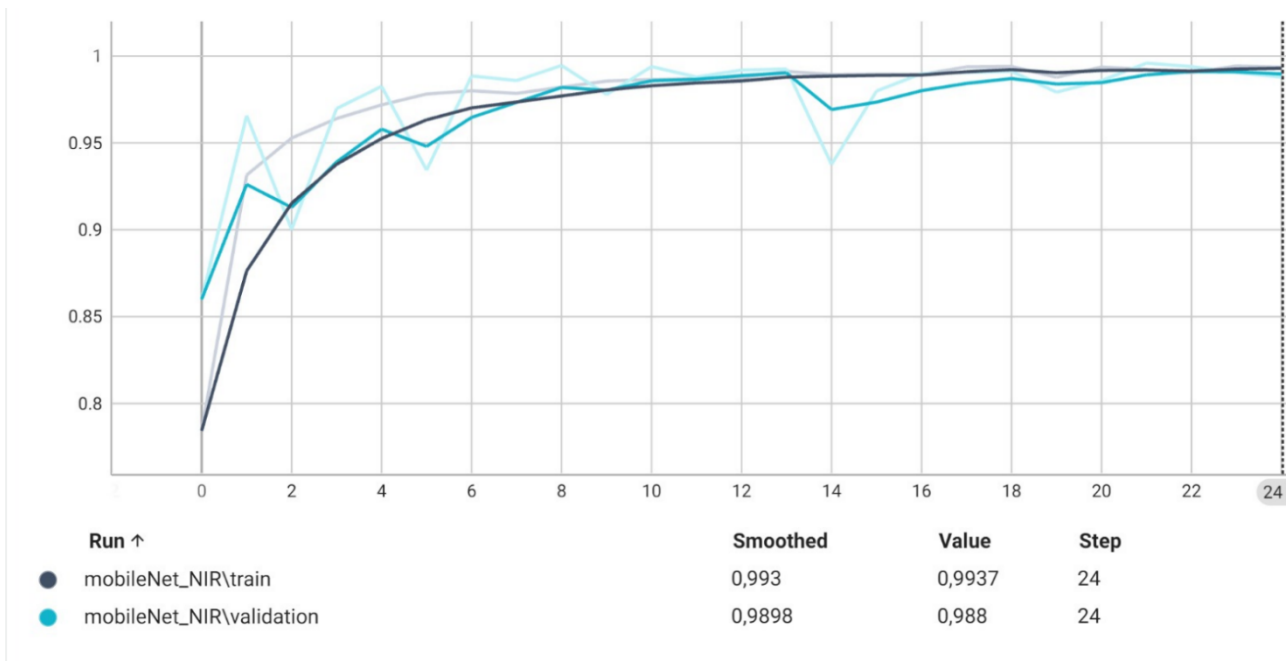
Se generó una matriz de confusión para analizar el rendimiento del modelo en términos de falsos positivos, falsos negativos y clasificaciones correctas para cada clase.

### **3.4 Selección del modelo final**

El modelo final seleccionado fue el MobilenetV1 de la arquitectura Feedforward. Se seleccionó este modelo debido a la métrica de accuracy obtenida para el espectro visible y NIR.

### 3.4.1 Criterios utilizados para seleccionar el modelo final

El criterio principal fue el del accuracy, ya que es un indicador general del rendimiento del modelo y, en este caso, los altos porcentajes de accuracy en ambos conjuntos de datos (98.39% en RGB y 99.37% en NIR) sugieren que el modelo es capaz de clasificar con éxito la mayoría de las imágenes de manzanas. Además, un modelo altamente preciso puede reducir los costos y mejorar la eficiencia operativa al minimizar los falsos positivos y falsos negativos.



**Figura 3.2 Curva de accuracy de modelo feedforward MobilenetV1 NIR**

La Figura 3.2 representa la curva de aprendizaje con el que se entrenó el modelo MobileNetV1 con arquitectura Feedforward e imágenes NIR de entrada. Las curvas de color azul y celeste representan el accuracy de los datos de entrenamiento y validación respectivamente. Se puede apreciar que ambas curvas convergen y permanecen paralelamente apegadas en casi todas las épocas, sumado a un alto porcentaje de accuracy en general. Esto significa que el modelo está

bien ajustado, tanto para el conjunto de entrenamiento como validación, así que el modelo puede generalizar bien a nuevos datos no vistos.

### 3.5 Evaluación del modelo

En la siguiente sección se evaluarán las arquitecturas según el resultado obtenido en el entrenamiento. El análisis se basará en las métricas de accuracy y el recall obtenido.

#### 3.5.1 Análisis de métricas de evaluación

Para 3 de las 4 arquitecturas de redes Feedforward se obtuvieron excelentes resultados, logrando obtener un accuracy superior al 95%. Esto permitió realizar un prototipo capaz de utilizar los resultados de clasificación de 3 modelos para obtener una mejor predicción final. El siguiente análisis tiene como objetivo proporcionar una visión detallada de cómo cada modelo se desempeñó en diferentes conjuntos y modalidades, permitiendo identificar fortalezas y debilidades específicas de

Modelos Feedforward	RGB		NIR	
	Train	Validation	Train	Validation
MobileNet	98,40%	98,26%	99,40%	98,80%
Resnet50	97,67%	96,10%	96,60%	95,79%
VGG19	33,32%	33,22%	33,42%	33,36%
Densenet121	97,57%	98,46%	95,57%	95,39%

**Tabla 3.3 Accuracy de modelos feedforward**

En la Tabla 3.3 se tiene el valor del *accuracy* correspondiente a cada arquitectura de redes *Feedforward* para el conjunto de datos de entrenamiento y validación en ambos espectros analizados.



Se puede apreciar que MobileNetV1, Resnet50 y DenseNet121 demostraron un alto rendimiento en el conjunto de entrenamiento RGB, alcanzando un *accuracy* de hasta 98.40% (MobileNet). Esto indica una capacidad sólida para aprender y generalizar patrones en las imágenes del espectro visible durante el entrenamiento. El *accuracy* en el conjunto de validación RGB se mantuvo alta, con hasta un 98,26% (MobileNet). Esto sugiere que el modelo logró generalizar bien a datos no vistos, mostrando consistencia con el conjunto de entrenamiento.

En el conjunto de entrenamiento NIR, MobileNet exhibió un *accuracy* de hasta 99,40% (MobileNet) indicando un fuerte rendimiento en la clasificación de defectos en manzanas basado en datos del espectro NIR. El *accuracy* en el conjunto de validación NIR fue de hasta 98,80% (MobileNet), lo que mostró al igual que en el conjunto de validación en RGB, una buena capacidad de generalización a imágenes no vistas durante el entrenamiento del espectro NIR [21].

Para el conjunto de entrenamiento y validación RGB y NIR de VGG19, se mostró un rendimiento significativamente inferior en ambos conjuntos. El *accuracy* en el rango del 30% sugiere dificultades significativas para aprender patrones y generalizar en comparación con los otros modelos.

En general, MobileNet y DenseNet121 presentaron las mejores métricas de *accuracy* en todos los conjuntos, tanto en el espectro visible como en el NIR. Resnet50 también mostró un rendimiento sólido, aunque ligeramente inferior al de MobileNet y DenseNet121. Por otro lado, VGG19 destacó por su notable rendimiento inferior en todos los conjuntos y modalidades.

### 3.5.2 Matriz de confusión y análisis de errores

Se generó la matriz de confusión en la Figura 3.3 con el conjunto de prueba que contiene un 10% del *dataset* y que fue seleccionado de manera aleatoria. Se utilizó el modelo entrenado con MobileNetV1 con imágenes NIR y con un *resize* (reajuste) de 960x830 a 224x224 de todas las imágenes. El conjunto de prueba consistió en 500 imágenes por clase. Se obtuvo un rendimiento sobresaliente en todas las categorías con una única instancia de error por clase.

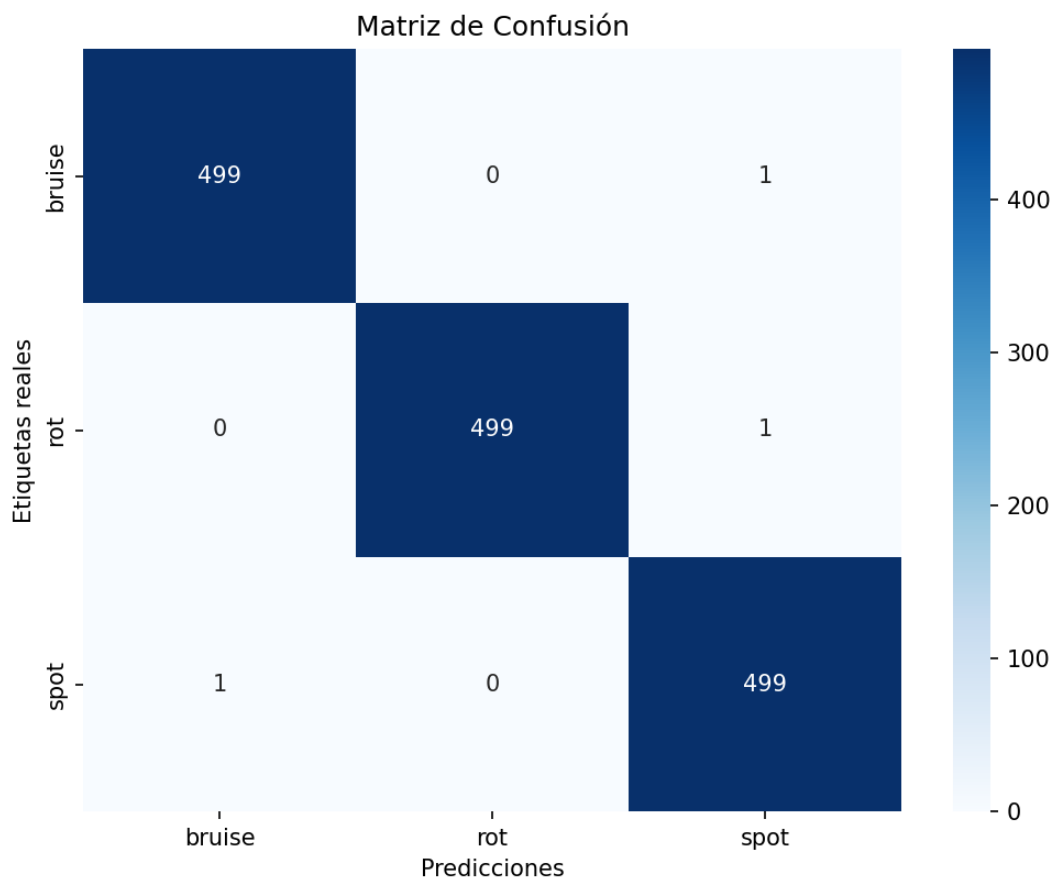


Figura 3.3 Matriz de confusión MobileNetV1 con imágenes NIR reajustadas

El modelo predijo correctamente 499 imágenes de la clase "bruise" y "spot", pero cometió un error al confundir una imagen de bruise y otra de spot con la clase "spot". También predijo correctamente 499 imágenes de la clase "spot", pero confundió una imagen de spot con la clase bruise.

### **3.6 Análisis de Errores**

A continuación, se analizarán en detalle los errores de predicción entre la clase "bruise" con "spot" y el error entre la clase "rot" y "spot". Se realizará el siguiente análisis con el fin de permitir una comprensión más profunda de los errores del modelo y ofrecer oportunidades para mejorar la capacidad de discriminación, especialmente en situaciones donde las características pueden ser más sutiles o confusas.

La confusión con la clase "spot" podría ser atribuida a la prevalencia de puntos negros en la imagen, un aspecto que el espectro NIR detecta en abundancia. Este factor de confusión podría haber llevado al modelo a asociarlo erróneamente con la clase "spot". Esta situación destaca la importancia de evaluar no solo los resultados finales sino también las características específicas que influyen en las decisiones del modelo.

Un análisis de refinamiento del modelo sería el de subrayar la necesidad de considerar la inclusión de características específicas que distingan claramente entre "bruise" y "spot". Además, ajustes adicionales en la ponderación de clases o técnicas específicas de mejora de la precisión en clases minoritarias podrían ser beneficiosos.

Dada la disponibilidad de manzanas y de una alta presencia de puntos negros en estas, es entendible que todos estos errores de predicción estén involucrados con la clase “spot”. Además, al usar las imágenes de manzanas con golpes del día 1, era poco notoria la marca del golpe, por lo que podía existir una presencia de puntos negros en mayor proporción que la del propio golpe

### **3.7 Interpretación de resultados**

El modelo VGG19 exhibió el menor nivel de precisión (accuracy) tanto en el conjunto de entrenamiento como en el de validación, mostrando una disminución significativa de más del 50% en comparación con otros modelos. A pesar de haber sido seleccionado por su eficacia en análisis similares de defectos en frutas [13], no logró desempeñarse de manera adecuada bajo la configuración de sensibilidad de pesos, regularización e hiperparámetros que fueron generalizados para todos los modelos. Se ha observado que este modelo destaca en situaciones específicas, como la detección de golpes individuales, pero su eficacia se ve disminuida al abordar la detección de otros tipos de defectos.

La dimensión de las imágenes en el conjunto de datos también influyó en la precisión del modelo final, ya que se empleó un número reducido de lotes de imágenes para el entrenamiento, la validación y las pruebas de los modelos. Por esta razón, el modelo final (MobileNetV1 NIR) fue entrenado con imágenes reajustadas a un tamaño de 224x224 píxeles para analizar posibles variaciones en el accuracy de predicción del conjunto de prueba y para medición de la curva de accuracy durante en el entrenamiento y validación. Los resultados fueron positivos para ambos casos, por lo que se consideró el modelo final reajustado por optimización de recursos durante la predicción del modelo en el prototipo final.

## **Capítulo 4**

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

En un contexto agrícola, la identificación precisa de defectos en las manzanas tiene implicaciones económicas significativas, y un modelo altamente preciso contribuye a la optimización de los recursos.

La investigación sobre la clasificación de defectos en manzanas mediante el uso de redes neuronales convolucionales (CNN) ha proporcionado valiosas percepciones y resultados significativos. Al analizar los datos de rendimiento de los modelos MobileNet, ResNet50, VGG19 y DenseNet121, se destacan varios puntos clave que influyen en la eficacia de la clasificación de defectos en manzanas.

Uno de estos es el rendimiento de los modelos Feedforward para MobileNetV1, DenseNet121 y ResnetNet50, el cual fue alto para el conjunto de entrenamiento y validación. Esta consistencia en el rendimiento entre ambos conjuntos indica que los modelos no sufren de sobreajuste significativo. La precisión más alta en el conjunto de entrenamiento NIR sugiere que los modelos tienen éxito en aprender patrones específicos del espectro NIR.

Uno de los hallazgos más destacados de este estudio es la trascendental importancia del espectro NIR en el proceso de clasificación. Los modelos entrenados con datos del espectro NIR exhibieron un accuracy sustancialmente más alto tanto en los conjuntos de entrenamiento como de validación. Este resultado subraya la capacidad única del espectro NIR para revelar patrones específicos relacionados con los defectos en las manzanas.

El accuracy más alto en el conjunto de entrenamiento NIR indica que los modelos fueron capaces de aprender y generalizar de manera efectiva patrones específicos en las imágenes del espectro NIR. Esto sugiere que la información del espectro NIR es fundamental para discernir características sutiles asociadas con defectos en las manzanas, esenciales para una clasificación precisa.

Se observó que las arquitecturas de redes neuronales MobileNet, DenseNet121 y ResNet50 destacaron por su robusto rendimiento en comparación con VGG19. La eficiencia de MobileNet, especialmente en conjuntos de entrenamiento y validación RGB, respalda su idoneidad para aplicaciones en tiempo real o dispositivos móviles con restricciones computacionales.

De VGG19 se esperaba un rendimiento que superara el 70% de accuracy por su simplicidad y estructura uniforme, donde se utilizan capas convolucionales de tamaño pequeño [24]. Sin embargo, mostró un rendimiento inferior en todos los conjuntos y modalidades, lo que destaca la influencia crítica de la elección de la arquitectura en la capacidad de aprendizaje del modelo.

En resumen, este estudio resalta la trascendental contribución del espectro NIR en la clasificación de defectos en manzanas y destaca la importancia de la elección de la arquitectura del modelo. Las conclusiones derivadas de este trabajo no solo tienen implicaciones directas en la industria agrícola, sino que también ofrecen valiosas perspectivas para futuras investigaciones en el campo de la visión por computadora aplicada a la agricultura.

## 4.2 Recomendaciones

### **Recomendaciones referentes a la adquisición de imágenes**

- Dado que la calidad de la cámara y el tamaño de imagen presentaron desafíos, se recomienda considerar la adquisición de una cámara con mayor resolución. Esto puede mejorar significativamente la calidad de las imágenes capturadas, lo que beneficiaría el accuracy del modelo al proporcionar detalles más finos, impactando positivamente en el rendimiento del modelo.
- Realizar la adquisición de imágenes con cámaras diseñadas específicamente para aplicaciones de clasificación de productos agrícolas. Estas cámaras suelen contar con características optimizadas para capturar detalles importantes para la clasificación de defectos.
- Explorar técnicas avanzadas de reducción de tamaño de imagen que minimicen la pérdida de información crucial para la clasificación. Esto podría incluir métodos de compresión sin pérdida o técnicas específicas de preprocesamiento para mantener la información relevante.

### **Recomendaciones referentes al entrenamiento de modelos**

- Considerar la exploración de arquitecturas de redes neuronales más eficientes, como MobileNetV1, que están diseñadas para funcionar de manera efectiva con restricciones de recursos, como lotes pequeños y tamaños de imagen reducidos.
- Realizar experimentos con diferentes tamaños de lote (batch sizes) para evaluar el equilibrio entre la eficiencia del entrenamiento y la calidad del modelo. Esto podría incluir la experimentación con tamaños de lote mayores una vez que se aborden los desafíos relacionados con el tamaño de la imagen.

Además de las recomendaciones anteriores, hay que considerar que, aunque la arquitectura siamesa no dio los resultados deseados en este estudio, no significa que deba



descartarse por completo. Los resultados insatisfactorios pueden atribuirse tanto a la falta de refinamiento en las siluetas de las imágenes como a las limitaciones en la adquisición de las imágenes utilizadas en el estudio.

Para futuras investigaciones, se recomienda considerar las siguientes acciones:

- Adquisición de imágenes utilizando equipos y cámaras de mayor calidad y resolución. Específicamente para la captura de imágenes en el espectro infrarrojo cercano (NIR), se recomienda utilizar cámaras diseñadas para captar en ese rango espectral. Esto permitirá obtener imágenes más precisas y consistentes en el rango de interés, lo cual es fundamental para aplicaciones que requieren un análisis detallado en el espectro infrarrojo cercano.
- Explorar técnicas de preprocesamiento adicionales para mejorar la calidad de las siluetas. Se sugiere considerar técnicas de segmentación precisa de las regiones de interés y algoritmos de suavizado. Esto permitirá obtener un conjunto de datos de siluetas más precisas y relacionadas con los defectos correspondientes.

Estas acciones adicionales pueden contribuir a mejorar el rendimiento y los resultados de la arquitectura siamesa en futuros estudios. Es importante recordar que la mejora en la adquisición de imágenes y el refinamiento de las siluetas son aspectos clave para obtener resultados más satisfactorios en aplicaciones de visión por computadora.

# Referencias

- [1] «El estado mundial de la agricultura y la alimentación 2019».
- [2] «Capítulo 2. Preparación para el mercado». Accedido: 9 de octubre de 2023. [En línea]. Disponible en: <https://www.fao.org/3/Y4893S/y4893s05.htm>
- [3] United Nations Economic Commission for Latin America and the Caribbean, Brechas y Transformaciones La Evolución del Empleo Agropecuario en América Latina. en ECLAC Books. UN, 2017. doi: 10.18356/daa2f77a-es.
- [4] «Guía de Defectos en WA 38 | WSU Tree Fruit | Washington State University». Accedido: 23 de octubre de 2023. [En línea]. Disponible en: <https://treefruit.wsu.edu/guia-de-defectos-en-wa-38/>
- [5] S. Arunachalam, H. H. Kshatriya, y M. Meena, «Identification of Defects in Fruits Using Digital Image Processing», Int. J. Comput. Sci. Eng., vol. 6, n.o 10, pp. 637-640, oct. 2018, doi: 10.26438/ijcse/v6i10.637640.
- [6] «FAOSTAT». Accedido: 9 de octubre de 2023. [En línea]. Disponible en: <https://www.fao.org/faostat/es/#data/QI>
- [7] «Revolución tecnológica e inclusión social: reflexiones sobre desafíos y oportunidades para la política social en América Latina».
- [8] W. Vásquez-Castillo, M. Racines-Oliva, P. Moncayo, W. Viera, y M. Seraquive, «Calidad del fruto y pérdidas poscosecha de banano orgánico (Musa acuminata) en el Ecuador», Enfoque UTE, vol. 10, n.o 4, pp. 57-66, 2019.
- [9] B. Ag, «Basler ace User's Manual for GigE Cameras».
- [10] «Segment Anything | Meta AI». Accedido: 24 de octubre de 2023. [En línea]. Disponible en: <https://segment-anything.com/>

[11] Ultralytics, «SAM (Segment Anything Model) ». Accedido: 24 de octubre de 2023. [En línea]. Disponible en: <https://docs.ultralytics.com/models/sam>

[12] J. D. Arango, B. Staar, A. M. Baig, y M. Freitag, «Quality control of apples by means of convolutional neural networks - Comparison of bruise detection by color images and near-infrared images», *Procedia CIRP*, vol. 99, pp. 290-294, 2021, doi: 10.1016/j.procir.2021.03.043.

[13] R. Pacheco, P. González, L. Chuquimarca, B. Vintimilla, y S. Velastin, «Fruit Defect Detection Using CNN Models with Real and Virtual Data»: en *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2023, pp. 272-279. doi: 10.5220/0011679800003417.

[14] R. Pourdarbani, S. Sabzi, M. Dehghankar, M. H. Rohban, y J. I. Arribas, «Examination of Lemon Bruising Using Different CNN-Based Classifiers and Local Spectral-Spatial Hyperspectral Imaging», *Algorithms*, vol. 16, n.o 2, p. 113, feb. 2023, doi: 10.3390/a16020113.

[15] D. Unay, B. Gosselin, O. Kleynten, V. Leemans, M.-F. Destain, y O. Debeir, «Automatic grading of Bi-colored apples by multispectral machine vision», *Comput. Electron. Agric.*, vol. 75, n.o 1, pp. 204-212, ene. 2011, doi: 10.1016/j.compag.2010.11.006.

[16] J. Li, X. Rao, y Y. Ying, «Detection of common defects on oranges using hyperspectral reflectance imaging», *Comput. Electron. Agric.*, vol. 78, n.o 1, pp. 38-48, ago. 2011, doi: 10.1016/j.compag.2011.05.010.

[17] J. F. I. Nturambirwe and U. L. Opara, “Machine learning applications to nondestructive

defect detection in horticultural products,” *Biosystems Engineering*, vol. 189, pp. 60–83, Jan. 2020, doi: 10.1016/j.biosystemseng.2019.11.011.

[18] Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. Fürnkranz, J and Joachims, T (eds.), Proceedings of the 27th International Conference on Machine Learning (ICML-10) 2010: 807-814.

[19] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[20] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778).

[22] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, No. 2).

[23] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[24] L. E. Chuquimarca, B. X. Vintimilla, and S. A. Velastin, "Banana ripeness level classification using a simple cnn model trained with real and synthetic datasets." in VISIGRAPP (5: VISAPP), 2023, pp. 536–543.

# APÉNDICES

<b>Apéndice A: Detalles de cámara NIR utilizada</b>	
<b>Componente</b>	<b>Detalles</b>
Cámara	Basler ace acA1300-60gmNIR
Marca	Basler
Modelo	Ace acA1300-60gmNIR
Resolución	1280 x 1024 píxeles
Tipo de sensor	CMOS
Formato de imagen	RAW, JPEG
Longitud de onda	NIR (Infrarrojo cercano)
Tipo de montura	C-Mount
Velocidad máxima de cuadro	60 fps
Dimensiones físicas	29 mm x 29 mm x 29 mm
Peso	46 g
Interfaz	GigE

La cámara Basler ace acA1300-60gmNIR fue utilizada durante la investigación para la adquisición de imágenes en el rango del espectro infrarrojo cercano (NIR). Se configuró para capturar imágenes en formato RAW con una resolución de 1280 x 1024 píxeles y una velocidad máxima de cuadro de 60 fps. Se montó utilizando una montura estándar C-Mount y se conectó a la computadora a través de una interfaz GigE para la transferencia de datos.

<b>Apéndice B: Detalles de cámara RGB utilizada</b>	
<b>Componente</b>	<b>Detalles</b>
Cámara	MER2-160-227U3C
Marca	DAHENG
Modelo	MER2-160-227U3C
Resolución	1280 x 960 píxeles
Tipo se sensor	CMOS
Formato se imagen	RAW, JPEG
Longitud de onda	Visible (RGB)
Tipo de montura	C-Mount
Velocidad máxima de cuadro	227 fps
Dimensiones físicas	29 mm x 29 mm x 29 mm
Peso	50 g
Interfaz	USB 3.0

La cámara DAHENG MER2-160-227U3C fue utilizada durante la investigación para la adquisición de imágenes en el espectro visible (RGB). Se configuró para capturar imágenes en formato RAW con una resolución de 1280 x 960 píxeles y una velocidad máxima de cuadro de 227 fps. Se montó utilizando una montura estándar C-Mount y se conectó a la computadora a través de una interfaz USB 3.0 para la transferencia de datos.