Facultad de Ingeniería y Tecnología Informática

Ingeniería Informática

Prototipo de Visión Artificial para la Clasificación de Manzanas mediante Imágenes Digitales

Trabajo Final de Carrera

Alumna: Chiara Brunella Tanzi

ID UB: 000-17-3192

Tutor: Ing. Carlos Alberto Esteves

Director de Carrera: Ing. Sergio Omar Aguilera

Agradecimiento

Antes de comenzar con el resumen de mi Trabajo Final de Carrera, quería tomarme un momento para agradecer a todas las personas que estuvieron conmigo durante estos seis largos años. A lo largo de este camino hubo muchos momentos buenos y también otros no tan buenos, pero nunca dejaron de acompañarme ni de confiar en mí.

En primer lugar, quiero agradecer a mi familia, que estuvo presente durante toda la carrera y especialmente en este último tramo. Incluso antes de empezar, cuando todavía no sabía qué carrera elegir y tenía muchas dudas sobre mi futuro, ellos me acompañaron, me escucharon y confiaron en mí. A lo largo de todos estos años, incluso cuando yo misma dudaba o cuando las cosas no salían como esperaba, ellos nunca me dejaron bajar los brazos y siempre me impulsaron a seguir adelante. Espero que sepan lo importantes que son para mí.

También quiero dedicar unas palabras a mi papá, porque esta tesina, de alguna manera, también nació gracias a él. Cuando estaba buscando una idea para desarrollar este trabajo, fue él quien me acercó esta problemática y me ayudó a conseguir las pruebas necesarias, además de hacer posible la visita a Kleppe, que fue fundamental para entender mejor el tema y darle más sentido al proyecto.

Quiero agradecer también a Carlos, mi tutor, por acompañarme en el desarrollo de la tesina, por sus devoluciones, su tiempo y su ayuda para ordenar, corregir y mejorar el trabajo en cada etapa.

Por último, quiero agradecer a mis compañeros, con quienes compartí tantos años de cursada, esfuerzo y nervios. Pasar noches estudiando, preparando entregas o terminando proyectos me enseñó no solo contenidos académicos, sino también a trabajar bajo presión y en equipo. También me llevo recuerdos y anécdotas divertidas que hicieron más llevadero el camino y que nunca me los voy a olvidar. Todo este recorrido hubiera sido mucho más difícil sin ellos.

Resumen

El presente Trabajo Final de Carrera propone el desarrollo de un prototipo de visión artificial orientado a la clasificación automatizada de manzanas para pequeños y medianos productores del Alto Valle del Río Negro, Argentina.

El problema abordado no radica en la ausencia de tecnología de clasificación en la industria existen líneas automatizadas de alta complejidad en grandes empresas exportadoras sino en la falta de acceso a herramientas de bajo costo para el segmento de productores que representa el 75% del total del sector y que aún realiza la clasificación manualmente a ojo. Esta dependencia del criterio humano introduce subjetividad y variabilidad que dificultan el cumplimiento de los estándares definidos por la normativa MERCOSUR GMC RES. 117/1996 [18].

Con el objetivo de demostrar la viabilidad técnica de una solución accesible, se implementaron y compararon dos enfoques metodológicos de complejidad creciente. El primero, basado en segmentación por color en el espacio HSV mediante la librería OpenCV, alcanzó una precisión del 6,2% sobre el conjunto de validación. El segundo, basado en Transfer Learning con la arquitectura MobileNetV2 pre entrenada en ImageNet e implementado con el framework PyTorch, alcanzó el 80% de precisión general, clasificando correctamente el 100% de las manzanas de Categoría III y Descarte, y el 80% de las manzanas Extra. Los extremos comerciales más críticos, Extra (sin defectos) y Descarte (no apta para comercialización), obtuvieron en conjunto una precisión promedio del 90%.

El sistema analiza múltiples fotografías tomadas desde distintos ángulos de una misma manzana, promedia las predicciones individuales y produce una clasificación final en alguna de las cuatro categorías definidas por la normativa: Extra, Categoría II, Categoría III y Descarte. El costo estimado de implementación del hardware requerido no supera los USD 40, utilizando en su mayoría equipamiento ya disponible en el entorno del pequeño productor.

Los resultados obtenidos validan parcialmente la hipótesis planteada, superando el umbral del 70% de precisión general. Las categorías con mayor impacto comercial (Extra y Descarte) alcanzaron en conjunto el 90% de precisión. Las limitaciones identificadas, principalmente el tamaño reducido del dataset (111 imágenes) y la dificultad en Categoría II, son consistentes con el alcance prototípico del trabajo.

1. Introducción

1.1 Formulación del problema

El Alto Valle del Río Negro concentra la mayor producción de manzanas de Argentina, con aproximadamente 17.000 hectáreas cultivadas y alrededor de 4.000 productores en las provincias de Río Negro y Neuquén. Según datos del SENASA [17], en Río Negro operan 1.359 productores de manzanas registrados, de los cuales el 75% posee menos de 20 hectáreas y representa apenas el 28% de la superficie total cultivada.

Este segmento de pequeños y medianos productores carece en su mayoría de acceso a tecnología automatizada de clasificación. A diferencia de las grandes empresas exportadoras como Kleppe S.A., con sede en Cipolletti, que opera sistemas de clasificación óptica industrial de última generación. Los pequeños productores dependen casi exclusivamente de la inspección visual manual realizada por operarios para determinar la calidad de su fruta según las categorías establecidas por la normativa MERCOSUR GMC RES. 117/1996.

Esta dependencia del criterio humano introduce subjetividad y variabilidad en la clasificación, dificultando el cumplimiento uniforme de los estándares de calidad. Una clasificación incorrecta puede derivar en rechazos en mercados de exportación, pérdida de valor comercial de la fruta y dificultades para garantizar la trazabilidad del producto. La falta de estandarización también compromete el cumplimiento de normas internacionales de exportación, aspecto crítico para la sustentabilidad económica del sector.

1.2 Justificación del trabajo

La investigación se justifica en la necesidad de desarrollar una herramienta de clasificación objetiva, reproducible y de bajo costo, accesible para pequeños productores sin requerir infraestructura industrial. El avance de las técnicas de visión artificial y aprendizaje profundo, en particular el Transfer Learning, ha hecho posible construir sistemas de clasificación de imágenes con alta precisión utilizando datasets reducidos y hardware convencional.

Desde el punto de vista académico, el trabajo integra conocimientos de procesamiento de imágenes, aprendizaje automático, normativa de calidad alimentaria e ingeniería de software, constituyendo un caso de aplicación concreta de la inteligencia artificial a un problema productivo regional de relevancia económica y social.

1.3 Objetivo general y específicos

Desarrollar un prototipo de visión artificial que, a partir de múltiples fotografías de cada manzana tomadas desde distintos ángulos, clasifique las frutas en las categorías de calidad definidas por la normativa MERCOSUR GMC RES. 117/1996, demostrando la viabilidad técnica de una solución de bajo costo para pequeños productores del Alto Valle del Río Negro.

Objetivos específicos:

- Diseñar un protocolo de captura de imágenes reproducible y estandarizado para la adquisición de fotografías de manzanas desde múltiples ángulos.
- Implementar y evaluar un método de clasificación basado en segmentación por color HSV utilizando la librería OpenCV.
- Implementar y evaluar un método de clasificación basado en Transfer Learning con la arquitectura MobileNetV2.
- Evaluar el desempeño de ambos métodos mediante métricas de clasificación como precisión, recall, F1-score y matriz de confusión.
- Comparar los métodos desarrollados en términos de desempeño predictivo, interpretabilidad, limitaciones técnicas y viabilidad práctica.
- Estimar el costo de implementación del prototipo propuesto, considerando su posible aplicación en contextos productivos de bajo presupuesto.
- Documentar las principales limitaciones identificadas durante el desarrollo y proponer líneas de mejora para futuras etapas del proyecto.

1.4 Hipótesis de trabajo

La implementación de un prototipo de visión artificial, capaz de capturar múltiples imágenes de cada manzana desde diferentes ángulos y procesarlas mediante técnicas de aprendizaje automático, permitirá clasificar las frutas en las categorías de calidad definidas por la normativa MERCOSUR con un nivel de precisión comparable o superior al obtenido mediante la inspección manual tradicional realizada por operarios humanos.

Las métricas de validación de la hipótesis son:

- Precisión general mayor al 70% sobre el conjunto de validación.
- Clasificación correcta del 90% de los casos extremos (Extra y Descarte), siendo Extra la categoría de mayor calidad sin defectos y Descarte la no apta para comercialización.
- Costo de implementación de hardware inferior a USD 40.

1.5 Metodología

El trabajo adopta un enfoque experimental basado en prototipado iterativo. Se desarrollaron dos métodos de clasificación de complejidad creciente, evaluando cada uno sobre el mismo dataset de validación para permitir una comparación objetiva.

El dataset fue construido con fotografías propias tomadas bajo condiciones controladas de iluminación y fondo, con manzanas previamente clasificadas por un experto para garantizar la veracidad de las etiquetas. La evaluación de los modelos se realizó mediante métricas estándar de clasificación: precisión (precision), sensibilidad (recall), F1-score y accuracy general, calculadas sobre un conjunto de validación separado que no fue utilizado en el entrenamiento.

1.6 Alcance del trabajo

El presente trabajo se limita al desarrollo de un prototipo experimental orientado a demostrar la viabilidad técnica del sistema, no a la construcción de un producto industrial. La captura de imágenes se realiza de forma manual, utilizando la cámara de una computadora convencional, bajo condiciones de iluminación y fondo controladas.

El sistema evalúa únicamente defectos superficiales visibles en las fotografías. Defectos internos o condiciones que requieren análisis no visual quedan fuera del alcance. La clasificación se aplica a manzanas individuales, adaptando los criterios de la normativa MERCOSUR, que originalmente se define para lotes, a una evaluación por fruto.

1.7 Organización del documento

El presente trabajo se organiza de la siguiente manera:

- Marco Teórico: fundamentos de visión artificial, redes neuronales convolucionales, Transfer Learning y normativa de referencia.
- Análisis del Problema: situación actual del sector, caso Kleppe S.A. y comparación con soluciones comerciales.
- Diseño del Sistema: arquitectura, protocolo de captura y construcción del dataset.
- Desarrollo del Prototipo: implementación de ambos métodos.
- Resultados y Evaluación: métricas, comparación y validación de hipótesis.
- Conclusiones: hallazgos, limitaciones y proyecciones futuras.
- Bibliografía, Anexo y Glosario.

2. Marco teórico

El presente capítulo reúne los fundamentos teóricos que sustentan el desarrollo del prototipo propuesto. En primer lugar, se aborda la aplicación de la visión artificial en el ámbito agroalimentario. Luego, se presentan los conceptos centrales de las redes neuronales convolucionales y del Transfer Learning aplicados a la clasificación de frutas. A continuación, se contextualiza el trabajo mediante una revisión comparativa de antecedentes relevantes y, finalmente, se expone la normativa MERCOSUR utilizada como referencia para la definición de las categorías de calidad.

2.1 Visión artificial aplicada a la agricultura

2.1.1 Concepto y alcance de la visión artificial

La visión artificial, también denominada visión por computadora o computer vision, es un área de la inteligencia artificial orientada al desarrollo de sistemas capaces de interpretar y analizar información visual proveniente del mundo real. Mediante técnicas de procesamiento digital de imágenes, extracción de características y clasificación, estos sistemas pueden identificar objetos, detectar defectos, estimar dimensiones y apoyar la toma de decisiones a partir de imágenes o secuencias de video [1].

Desde el punto de vista computacional, una imagen digital puede entenderse como una representación matricial de información visual, en la que cada posición almacena valores de intensidad o color. En las imágenes a color, esta información suele organizarse en tres canales, típicamente rojo, verde y azul (RGB). Sobre esa representación se aplican operaciones de transformación, filtrado y análisis destinadas a extraer información útil para una tarea específica, como la detección de defectos o la clasificación de calidad [2][3].

2.1.2 Aplicaciones en la industria agroalimentaria

La aplicación de la visión artificial en el sector agrícola y agroalimentario ha experimentado un crecimiento sostenido desde la década de 1990, impulsado por la reducción de costos del hardware de captura y el avance de los algoritmos de procesamiento. Según la revisión sistemática de Rojas Santelices [1], que analizó la literatura científica publicada entre 2018 y 2024 siguiendo la metodología PRISMA, las áreas de aplicación más frecuentes son la detección de defectos post-cosecha, la clasificación por madurez, la estimación de rendimiento, la detección de plagas y enfermedades, y el monitoreo del crecimiento.

En el contexto específico de la clasificación de frutas, la visión artificial ofrece ventajas determinantes respecto a la inspección manual: objetividad, ya que el sistema aplica siempre los

mismos criterios sin variabilidad entre operarios ni fatiga acumulada; velocidad, dado que puede procesar decenas de frutas por segundo en líneas industriales; y trazabilidad, puesto que registra automáticamente el resultado de cada clasificación para su auditoría posterior [4].

En Argentina, el contexto del Alto Valle del Río Negro presenta una necesidad específica: el 75% de los productores posee menos de 20 hectáreas y carece de acceso a las líneas automatizadas de clasificación óptica que utilizan las grandes empresas exportadoras. Para este segmento, la clasificación manual sigue siendo la práctica dominante, con las limitaciones de subjetividad y variabilidad que implica. El desarrollo de herramientas de visión artificial de bajo costo y fácil implementación representa una oportunidad concreta de mejora en la cadena productiva de estos pequeños productores [16].

2.1.3 Procesamiento de imágenes: etapas fundamentales

El funcionamiento de un sistema de visión artificial aplicado a la clasificación de frutas puede describirse como una secuencia de etapas complementarias.

En primer lugar, se realiza la adquisición de la imagen mediante una cámara digital, webcam o sensor especializado. En esta etapa, las condiciones de iluminación, la distancia de captura y el ángulo de observación influyen directamente en la calidad de los resultados posteriores.

A continuación, se lleva a cabo el preprocesamiento, que incluye operaciones destinadas a mejorar la calidad de la imagen, tales como la corrección de iluminación, la reducción de ruido, el ajuste de contraste y el redimensionado. En esta fase también puede incorporarse la conversión entre espacios de color, por ejemplo, de RGB a HSV.

Luego se aplica la segmentación, cuyo objetivo es separar la región de interés, en este caso la fruta, del fondo de la imagen. Esta separación puede basarse en umbrales de color, detección de contornos o técnicas de aprendizaje automático.

Una vez segmentada la región de interés, se procede a la extracción de características, es decir, a la obtención de descriptores numéricos que representan propiedades relevantes como el color, la textura, la forma o el área afectada por defectos.

Finalmente, se realiza la clasificación, etapa en la que la imagen o el fruto analizado se asigna a una categoría predefinida mediante un algoritmo de decisión, que puede consistir en reglas por umbral, clasificadores clásicos o redes neuronales.

2.2 Clasificación de manzanas mediante redes neuronales convolucionales

2.2.1 Del procesamiento clásico al aprendizaje profundo

Los primeros sistemas de clasificación de frutas por visión artificial se basaban en enfoques de procesamiento de imágenes clásico: extracción manual de características (histogramas de color, descriptores de textura, momentos geométricos) combinada con clasificadores como redes neuronales de una capa oculta, máquinas de vectores de soporte (SVM) o árboles de decisión. Estos métodos requerían que el investigador definiera explícitamente qué características extraer, lo que limitaba su generalización y exigía un proceso de ingeniería de características específico para cada problema.

En Colombia, Pencue y León-Téllez [11] desarrollaron uno de los primeros trabajos iberoamericanos sobre detección de defectos en frutas mediante procesamiento digital de imágenes, utilizando redes neuronales tipo perceptrón multicapas entrenadas con descriptores HSI sobre imágenes de naranjas Valencia. Los autores reportaron resultados prometedores, pero identificaron como limitación principal la dependencia de las condiciones de iluminación y la dificultad para generalizar a múltiples variedades.

Con la irrupción del aprendizaje profundo, en particular de las redes neuronales convolucionales (CNN) a partir de 2012, el paradigma cambió radicalmente: en lugar de definir manualmente las características, el modelo las aprende directamente de los datos durante el entrenamiento. Esto permitió saltos cualitativos de precisión en tareas de clasificación de imágenes que anteriormente parecían intratables [6].

2.2.2 Arquitectura de las redes neuronales convolucionales

Una red neuronal convolucional, o CNN, es un tipo de red neuronal profunda diseñada para procesar datos con estructura de cuadrícula, como las imágenes digitales. Su arquitectura combina distintos tipos de capas que permiten extraer, resumir y combinar información visual de manera jerárquica, desde patrones simples hasta representaciones más complejas relevantes para la clasificación final.

Capas convolucionales:

Son el componente fundamental de las CNN. Una capa convolucional aplica un conjunto de filtros aprendibles (también llamados kernels) sobre la imagen de entrada. Cada filtro es una pequeña matriz, típicamente de 3×3 o 5×5 píxeles, que se desliza sobre toda la imagen realizando el producto escalar entre sus pesos y los valores de los píxeles cubiertos. El resultado de esta operación para cada posición del filtro forma un mapa de activación (feature map), que indica en qué zonas de la imagen está presente la característica que detecta ese filtro. Las primeras capas detectan características simples como bordes horizontales, verticales y esquinas; las capas más profundas combinan estas características simples para detectar patrones complejos como texturas, manchas o formas características de defectos superficiales [6].

Capas de pooling (agrupamiento):

Se aplican generalmente entre capas convolucionales. Su función es reducir el tamaño espacial de los mapas de activación, conservando únicamente la información más relevante. La variante más utilizada es el max pooling: el mapa se divide en bloques no solapados (típicamente de 2×2) y de cada bloque se conserva el valor máximo. Esto produce una representación más compacta que reduce el costo computacional de las capas siguientes, hace al modelo más robusto ante pequeñas variaciones de posición del defecto en la imagen y actúa como regularizador al disminuir el número de parámetros [6].

Capas densas (fully connected):

Al final de la arquitectura convolucional, los mapas de activación se aplanan en un vector unidimensional que alimenta una o más capas densas. En una capa densa, cada neurona está conectada con todas las neuronas de la capa anterior. Estas capas combinan las características extraídas por la parte convolucional para producir la clasificación final. La última capa densa tiene tantas neuronas como categorías existen en el problema, y utiliza la función de activación Softmax para producir una distribución de probabilidades sobre las clases [5].

2.2.3 Trabajos en clasificación de manzanas con CNN

La manzana es uno de los productos agrícolas más estudiados en el campo de la visión artificial aplicada al control de calidad, debido a su importancia económica global y a la variedad de defectos superficiales definidos en las normativas de comercialización.

En el ámbito iberoamericano, Olguín-Rojas et al. desarrollaron un sistema de clasificación de manzanas mediante redes neuronales convolucionales, evaluando arquitecturas como LeNet5 y VGG16 sobre distintas variedades de manzana. Los autores reportaron resultados de alta exactitud, alcanzando hasta un 97% con la arquitectura de mejor desempeño. Este antecedente refuerza la viabilidad del uso de CNN para tareas de clasificación automática de frutas. [12]

En Ecuador, Aguilar-Alvarado y Campoverde-Molina [13] desarrollaron un clasificador de frutas basado en MobileNet de TensorFlow, entrenado sobre 13 categorías de frutas capturadas en diferentes ambientes. Su trabajo es especialmente relevante para el presente TFC porque comparte el uso de MobileNet como arquitectura base y el enfoque de clasificación con pocos recursos computacionales. Los autores concluyen que el modelo permite desarrollar sistemas autónomos con bajo costo computacional, lo que lo hace adecuado para contextos donde no se dispone de hardware especializado [13].

A nivel internacional, Fan et al. [14] desarrollaron un sistema de detección de manzanas defectuosas en línea de clasificación de 4 canales, combinando CNN con cámara comercial e iluminación LED. El sistema alcanzó una precisión del 96,5%, un recall del 100% y una especificidad del 92,9%, con capacidad de procesar 5 frutas por segundo. Li et al. (2021) reportaron una precisión del 95,33% en clasificación de calidad de manzanas mediante CNN entrenada con fotografías de 300 manzanas de diferentes variedades [14] [15].

2.3 Transfer Learning para clasificación con datasets reducidos

2.3.1 Fundamentos del Transfer Learning

El Transfer Learning (aprendizaje por transferencia) es una técnica de aprendizaje automático que consiste en reutilizar el conocimiento adquirido por un modelo entrenado en una tarea o dominio de origen para resolver una tarea o dominio objetivo diferente. El concepto fue formalizado por Pan y Yang [7] en su influyente surge publicado en IEEE Transactions on Knowledge and Data Engineering, donde establecieron el marco teórico del Transfer Learning y lo clasificaron según la relación entre dominio origen y dominio destino.

En el contexto de la clasificación de imágenes con redes neuronales profundas, el Transfer Learning opera de la siguiente manera: se parte de un modelo preentrenado en un dataset de gran escala, típicamente ImageNet, con 1,2 millones de imágenes y 1000 categorías, y se adapta para la tarea objetivo. La justificación teórica es que las primeras capas de una CNN preentrenada aprenden representaciones visuales de bajo nivel (detectores de bordes, gradientes, texturas simples) que son transferibles entre dominios visuales distintos, ya que estas características son universales a cualquier tipo de imagen [6] [7].

2.3.2 Estrategias de Transfer Learning

En términos generales, el Transfer Learning puede aplicarse mediante dos estrategias principales. La primera es la extracción de características, en la que las capas del modelo preentrenado se mantienen congeladas y se utilizan únicamente como extractor visual, entrenando solo el clasificador agregado al final. Esta opción resulta especialmente adecuada cuando el dataset objetivo es pequeño o muy similar al dataset de origen. La segunda estrategia es el fine-tuning, en la que además del clasificador final se descongelan algunas capas profundas del modelo pre entrenado y se ajustan con una tasa de aprendizaje reducida. Esto permite adaptar parcialmente las representaciones de alto nivel al nuevo dominio, preservando al mismo tiempo el conocimiento general adquirido durante el preentrenamiento.

2.3.3 MobileNetV2

MobileNetV2 [8] es una arquitectura CNN diseñada por Google para aplicaciones con restricciones de recursos computacionales. Sus principales innovaciones son los bloques de botella invertidos (inverted residual blocks) y las convoluciones separables en profundidad (depthwise separable convolutions). Las convoluciones separables factorizan la convolución estándar en dos operaciones más simples: una convolución en profundidad (que aplica un filtro separado a cada canal) y una convolución puntual (que combina los resultados). Esto reduce el número de operaciones matemáticas por un factor de 8 a 9 veces respecto a una convolución estándar, con una pérdida mínima de precisión [8].

Estas características hacen a MobileNetV2 especialmente adecuada para el presente trabajo: puede ejecutarse en una computadora convencional sin GPU de alto rendimiento para la inferencia (clasificación de manzanas nuevas), aunque el entrenamiento se realizó con GPU para acelerar el proceso. En el ámbito iberoamericano, Aguilar-Alvarado y Campoverde-Molina [13] validaron el uso de MobileNet para clasificación de frutas con bajos recursos computacionales, obteniendo resultados satisfactorios en 13 categorías de frutas.

2.4 Consideraciones teóricas sobre datasets reducidos

El dataset utilizado en el presente trabajo consta de 111 imágenes totales (91 de entrenamiento, 20 de validación), distribuidas en 4 categorías. Este tamaño es reducido en comparación con los datasets utilizados en investigaciones de gran escala. La siguiente justificación formal explica por qué resulta aceptable para el alcance prototípico del trabajo para los objetivos del trabajo y cómo se mitigaron sus limitaciones.

2.4.1 Transfer Learning como solución al problema de datos escasos

La técnica de Transfer Learning fue diseñada precisamente para escenarios con datos escasos. Los trabajos de Kornblith [9] demuestran que los modelos preentrenados en ImageNet transfieren representaciones útiles incluso cuando el dataset objetivo tiene pocas imágenes, ya que las características de bajo y medio nivel son independientes del dominio específico. Estudios con datasets de tan solo 50 a 100 imágenes por clase han demostrado resultados satisfactorios cuando se aplica Transfer Learning con fine-tuning.

En el contexto específico de frutas, el trabajo de Aguilar-Alvarado y Campoverde-Molina [13] validaron el uso de MobileNet con datasets de tamaño reducido, obteniendo resultados funcionales en clasificación de 13 categorías de frutas. Asimismo, Granados-Vega [4] desarrollaron un sistema de visión artificial de bajo costo para clasificación de limones con tan solo 90 muestras divididas en tres categorías, demostrando la viabilidad del enfoque con datasets pequeños cuando se aplican técnicas adecuadas de regularización.

2.4.2 Data augmentation como amplificación del dataset

El data augmentation es una técnica ampliamente aceptada para mitigar el problema de los datasets pequeños. Al generar versiones modificadas de cada imagen durante el entrenamiento (rotaciones, espejos, variaciones de brillo, zoom), se incrementa artificialmente la diversidad de los ejemplos que ve el modelo en cada época. Esto tiene dos efectos demostrados: reduce el sobreajuste (ya que el modelo no puede memorizar imágenes idénticas) y mejora la robustez ante variaciones del entorno real [10].

En el presente trabajo se aplicaron varias transformaciones de augmentación durante el entrenamiento, lo que permitió ampliar de manera efectiva la variabilidad del conjunto original de imágenes. Aunque esta estrategia no sustituye la necesidad de contar con más datos reales, sí mejora la capacidad del modelo para generalizar dentro del alcance experimental del prototipo.

2.4.3 Alcance prototípico del trabajo

El presente trabajo tiene como objetivo explícito demostrar la viabilidad técnica del sistema, no construir un modelo de producción industrial. En este contexto, un dataset de 111 imágenes es suficiente para validar la hipótesis planteada y obtener métricas representativas del comportamiento del sistema. La ampliación del dataset a 200 o más imágenes por categoría, con múltiples variedades de manzana y condiciones de iluminación variables, constituye la primera línea de trabajo futuro identificada en las conclusiones.

2.5 Comparación con trabajos relacionados

Con el fin de contextualizar el presente prototipo, resulta pertinente comparar sus características con antecedentes relevantes en clasificación de frutas mediante visión artificial. Debe tenerse en cuenta, sin embargo, que los trabajos relevados no son completamente homogéneos en cuanto a tamaño de dataset, condiciones de captura, arquitectura empleada y métrica principal reportada. Por ello, la comparación debe interpretarse como una contextualización general más que como una equivalencia experimental estricta.

Trabajo	Método	Dataset	Precisión	Observaciones
Pencue y León-Téllez — Colombia	RNA + HSI	Naranjas Valencia	~90%	Uno de los primeros trabajos iberoamericanos en detección de defectos en frutas por PDI.

Jiménez Ortiz — México	CNN (OpenCV + TensorFlow)	2.800 imgs manzanas	95,36% val.	Clasificación en tiempo real mediante video (92,25%). Código abierto.
Aguilar-Alvarado y Campoverde-Molinaet — Ecuador	MobileNet + TensorFlow	13 categorías frutas	Variable	Validación de MobileNet con pocos recursos computacionales para frutas iberoamericanas.
Fan — China	CNN + cámara comercial	Línea industrial	96,5%	Sistema en línea, 5 frutas/segundo. Recall 100% para defectos.
Li — China	CNN custom	300 manzanas	95,33%	Múltiples variedades. Comparación con SVM y Random Forest.
Granados-Vega — México	RNA + PDI	90 limones	Alta	Sistema de bajo costo. Validación con dataset pequeño (3 categorías).
Presente trabajo (2026) — Argentina	MobileNetV2 + TL (PyTorch)	111 imgs (4 cat.)	80%	Bajo costo, pequeños productores, Alto Valle Río Negro. Primer prototipo de este tipo para la región.

Tabla 2.1. Comparación con trabajos relacionados en clasificación de frutas mediante visión artificial.

El presente trabajo se diferencia de los antecedentes relevados en tres aspectos principales. En primer lugar, se sitúa en un contexto geográfico y productivo específico, el del Alto Valle del Río Negro, para el cual no se identificaron antecedentes equivalentes orientados a pequeños productores. En segundo lugar, adopta explícitamente un enfoque de bajo costo y accesibilidad, lo que condiciona tanto el protocolo de captura como la elección de la arquitectura. En tercer lugar, compara sobre un mismo dataset un enfoque clásico basado en reglas de color y un enfoque de aprendizaje profundo, lo que permite dimensionar con mayor claridad el beneficio relativo del modelo más complejo.

La diferencia entre la precisión obtenida en el presente trabajo y la reportada por antecedentes internacionales más cercanos al estado del arte puede explicarse principalmente por tres factores: el menor tamaño del dataset utilizado, las condiciones de captura no industriales y la

evaluación restringida a una única variedad de manzana. Estas diferencias son coherentes con el alcance prototípico declarado y no invalidan la relevancia del sistema desarrollado como prueba de viabilidad técnica.

2.6 Normativa de referencia: MERCOSUR GMC RES. 117/1996

2.6.1 Marco normativo

La clasificación de manzanas en el ámbito del MERCOSUR se rige por la Resolución GMC N° 117/1996, que establece el Reglamento Técnico para la Identidad y Calidad de la Manzana destinada al consumo humano. Esta normativa fue adoptada en Argentina mediante la Resolución SAGPyA N° 600/97 (Boletín Oficial, 2 de septiembre de 1997) y es de aplicación obligatoria para la comercialización e identificación del producto tanto en el mercado interno como para la exportación a los países miembros del MERCOSUR (SAGPyA, 1997).

2.6.2 Categorías de calidad

La normativa define cuatro categorías de calidad en función de los defectos presentes en el fruto, ordenadas de mayor a menor exigencia:

- Categoría Extra: frutas de calidad superior, prácticamente sin defectos visibles. El total de defectos no puede superar el 5% de las unidades de la muestra, con un máximo del 2% para defectos graves como machucamiento y heridas.
- Categoría II: frutas de buena calidad con defectos leves tolerables. Se admite hasta un 12% de unidades con defectos totales, con un máximo del 6% para defectos graves.
- Categoría III: frutas que no alcanzan la Categoría II pero cumplen los requisitos mínimos de comercialización. Se admite hasta un 20% de unidades con defectos totales, con un máximo del 10% para defectos graves.
- Descarte: frutas que superan los límites de Categoría II o presentan podredumbre, contaminación u otras condiciones que las hacen inaptas para el consumo humano.

2.6.3 Defectos clasificables

La normativa contempla distintos defectos que afectan la calidad comercial de la manzana. Entre ellos, los defectos graves tienen un rol central en la determinación de la categoría del fruto. La Tabla 2.2 resume los principales defectos graves considerados por la normativa de referencia.

Defecto	Descripción según normativa

Machucamiento	Lesión con deformación superficial sin rotura de epidermis, de origen mecánico (golpe, presión).
Herida abierta	Lesión con rotura de epidermis, independientemente de su tamaño.
Quemado de sol	Zona decolorada o necrosada por exposición excesiva a la radiación solar directa.
Lesión cicatrizada	Herida o daño superficial curado, con formación de tejido corchoso. Clasificada como grave o leve según su extensión.
Russeting (oxidación)	Manchas superficiales de color marrón claro o dorado, de origen fisiológico o ambiental.
Mancha de sarna	Lesiones causadas por el hongo Venturia inaequalis, de coloración oscura y textura rugosa.
Daño por granizo	Lesiones de forma circular producidas por impacto de granizo.

Tabla 2.2. Defectos graves clasificables según MERCOSUR GMC RES. 117/1996.

2.6.4 Adaptación al presente trabajo

La normativa MERCOSUR establece los límites de tolerancia en función del porcentaje de unidades defectuosas dentro de un lote de comercialización, no por fruto individual. El presente prototipo adapta esta clasificación a una evaluación individual de cada manzana, utilizando las mismas categorías como etiquetas de referencia. Esta adaptación es una simplificación válida en el contexto de un prototipo experimental, dado que el objetivo es demostrar la viabilidad técnica del sistema de clasificación visual, no replicar exactamente el proceso normativo de inspección de lotes.

Las manzanas utilizadas en el dataset fueron previamente clasificadas por el sistema industrial Spectrim de TOMRA en las instalaciones de Kleppe S.A., y verificadas por un operario experto de la empresa. Esto garantiza que las etiquetas del dataset son congruentes con los criterios normativos, aun cuando la evaluación individual del prototipo constituya una adaptación del proceso original de clasificación por lotes.

3. Análisis del Problema

3.1 Situación actual de la clasificación en el Alto Valle del Río Negro

El Alto Valle del Río Negro constituye la principal zona productora de manzanas de Argentina. Según datos del SENASA [17], en la Patagonia Norte operan 1.359 productores de manzanas registrados en la provincia de Río Negro y 183 en Neuquén, sobre una superficie total de aproximadamente 17.279 hectáreas cultivadas con esta especie.

La estructura productiva del sector presenta una marcada asimetría: el 75% de los productores posee menos de 20 hectáreas y concentra apenas el 28% de la superficie total cultivada, mientras que el 2,5% de los productores más grandes controla el 35% del área implantada [16]. Esta heterogeneidad no se limita a la superficie, también se extiende a la capacidad tecnológica y al acceso a herramientas de control de calidad.

De acuerdo con el Informe de Pera y Manzana (2021) del Ministerio de Agricultura, la clasificación de la fruta en el eslabón de empaque depende principalmente de la tecnología disponible en cada establecimiento. Las grandes empresas exportadoras han incorporado sistemas automatizados de clasificación óptica, mientras que en los establecimientos de menor escala la clasificación sigue realizándose en parte de forma manual. Esta situación implica una dependencia del criterio visual del operario, que introduce variabilidad e inconsistencias en la asignación de categorías conforme a la normativa MERCOSUR.

La crisis estructural del sector agudiza esta brecha. Según el mismo informe, entre 2013 y 2021 dejaron la producción 934 productores con la desaparición del 46% de los productores con menos de 10 hectáreas. La reducción de márgenes económicos limita aún más la posibilidad de inversión en tecnología de clasificación para los productores que permanecen en el sector.

3.2 Caso de estudio: Sistema Spectrim en Kleppe S.A.

Con el objetivo de contextualizar la brecha tecnológica existente entre grandes y pequeños productores, se realizó una visita técnica a las instalaciones de Kleppe S.A., una de las principales empresas frutícolas del Alto Valle del Río Negro. Kleppe S.A. es una empresa con sede en Cipolletti fundada en 1932, que cuenta con más de 3.000 empleados en temporada alta, tres plantas de empaque, cinco frigoríficos con capacidad para 40.000 pallets y aproximadamente 2.000 hectáreas propias. Exporta a más de 40 países en cuatro continentes bajo las certificaciones ISO 22.000, Global GAP y BRC Global Standards.

Durante la visita se observó en operación la línea de clasificación basada en el sistema Spectrim, desarrollado por TOMRA Food, fabricante noruego líder mundial en tecnología de clasificación óptica para la industria alimentaria. El sistema Spectrim clasifica las manzanas en primera instancia según sus imperfecciones superficiales y luego las calibra según defectos mayores y menores, incluyendo imperfecciones de la piel, daños por insectos, fruta de forma irregular, arañazos y abrasiones.

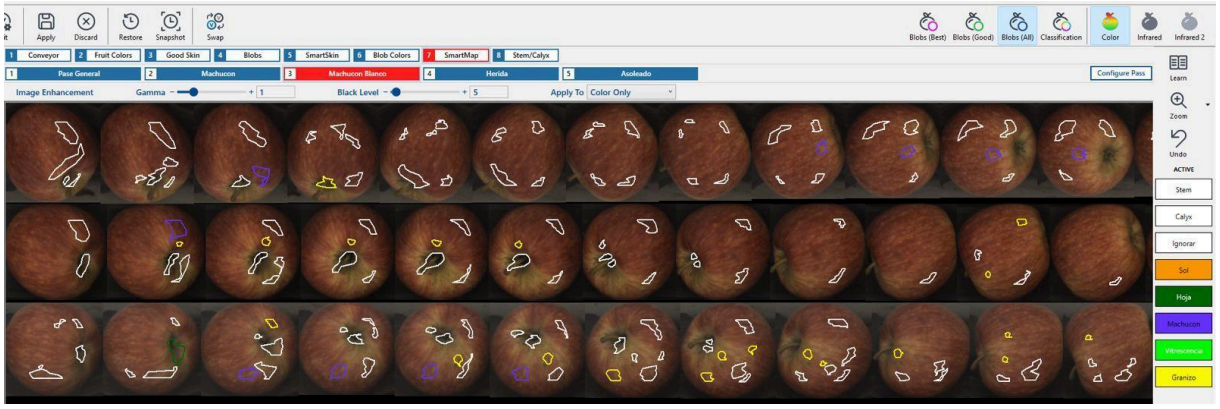


Figura 3.1. Interfaz del sistema Spectrim observada durante la clasificación de manzanas en Kleppe S.A.

La versión Spectrim con LUCAi®, incorporada en las líneas de empaque modernas, utiliza deep learning para la detección precisa de defectos, alcanzando la detección de más del 95% de las imperfecciones relacionadas con el pedúnculo y mitigando hasta el 99% de los defectos totales del pedúnculo. Esta tecnología opera de forma continua sobre líneas de varios carriles en paralelo, clasificando la fruta a velocidades industriales incompatibles con el equipamiento doméstico o de pequeña escala.

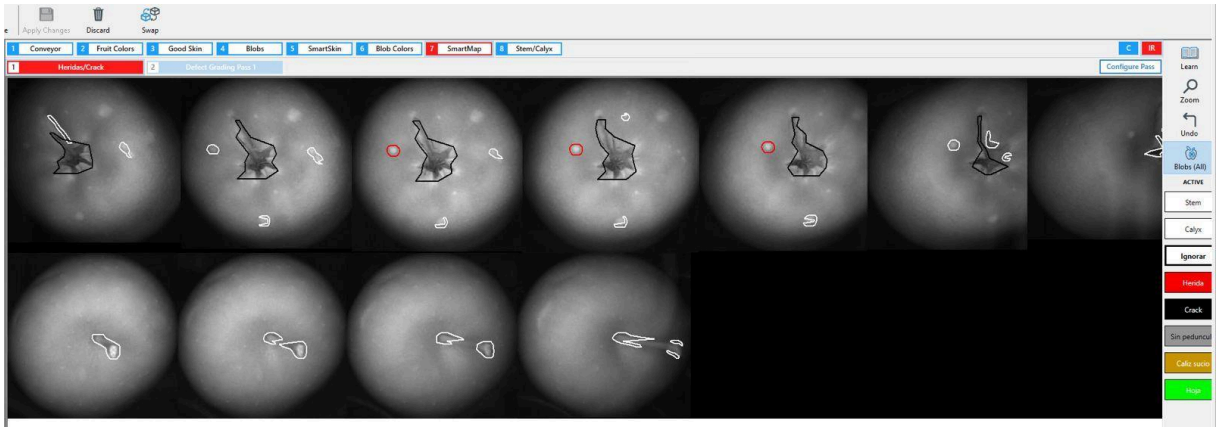


Figura 3.2. Ejemplo de visualización del sistema Spectrim para la detección de defectos superficiales en manzanas.

No obstante, los valores reportados por el fabricante corresponden a material comercial y a métricas específicas del sistema, por lo que no deben interpretarse automáticamente como equivalentes a la precisión global de clasificación observada en una planta real. Durante la visita técnica y en las conversaciones mantenidas con personal de la empresa, se mencionó que la performance operativa observada podía ubicarse en torno al 87–88%, valor que no necesariamente contradice la información del proveedor, ya que probablemente refiere a condiciones reales de

operación y a un criterio de evaluación más amplio que el utilizado en la comunicación comercial del producto.

La visita técnica permitió verificar empíricamente que el sistema Spectrim y el prototipo desarrollado en el presente trabajo comparten el mismo paradigma tecnológico, visión artificial con inteligencia artificial para clasificación de defectos por imagen, pero operan en escalas completamente distintas. Esta verificación refuerza la pertinencia del presente trabajo: si la tecnología de base es la misma, el desafío consiste en hacerla accesible a una escala compatible con los recursos del pequeño productor.

3.3 Comparación con soluciones comerciales existentes

La siguiente tabla resume las principales soluciones de clasificación automatizada disponibles en el mercado, comparadas con el prototipo desarrollado en el presente trabajo:

Sistema	Fabricante	Tecnología	Escala	Accesibilidad
Spectrim con LUCAi®	TOMRA (Noruega)	Deep learning, cámaras industriales multiespectrales	Industrial continua	Solo grandes empresas
Apples Sort 3	Unisorting (Italia)	Visión artificial + cámaras HD + análisis interno	Industrial	Solo grandes empresas
HDiA System	Quadra/Ser.mac (Italia)	IA en tiempo real, aprendizaje adaptativo	Industrial	Solo grandes empresas
Prototipo TFC (presente)	—	MobileNetV2, Transfer Learning, PyTorch	Individual por fotos	Pequeños productores

Tabla 3.1. Comparación diferentes soluciones comerciales

Las soluciones industriales requieren infraestructura de empaque especializada, instalación y calibración por técnicos certificados, y representan inversiones que oscilan entre cientos de miles y millones de dólares. Estas condiciones son incompatibles con la realidad económica del segmento de pequeños productores que representa el 75% del sector. El prototipo desarrollado en este trabajo no pretende competir en velocidad ni escala con estas soluciones, sino ofrecer una herramienta de verificación objetiva y reproducible accesible con equipamiento convencional.

3.4 Costo del sistema propuesto

Una de las características diferenciales del prototipo es su costo de implementación reducido. La siguiente tabla detalla los componentes necesarios y su costo estimado:

Componente	Costo estimado	Observación
Computadora personal	USD 0	Utiliza equipamiento ya disponible
Cámara (webcam o notebook)	USD 0	Utiliza equipamiento ya disponible
Iluminación LED constante	USD 10–30	Lámpara de escritorio estándar
Bandeja de fondo neutro	USD 2	Cartón o plástico de color uniforme
Software (Python + librerías)	USD 0	Todo el stack es open source y gratuito

TOTAL ESTIMADO	**USD 12–32**	**Sin contar hardware ya disponible**

Tabla 3.2. Costos del sistema

Para contextualizar esta cifra: una línea de clasificación industrial como el Spectrim representa una inversión de varios cientos de miles de dólares, sin considerar los costos de instalación, mantenimiento anual y capacitación de operarios especializados. La diferencia de escala, de dos a cuatro órdenes de magnitud, define claramente el segmento al que apunta el presente prototipo y justifica su desarrollo como herramienta complementaria al proceso productivo del pequeño productor.

3.5 Viabilidad en el campo real

Para evaluar la factibilidad de implementación del sistema en un contexto productivo real, se analizan los siguientes factores:

3.5.1 Condiciones de captura

El protocolo de captura definido, fondo oscuro uniforme, iluminación constante, distancia fija cámara-fruta, cuatro ángulos equidistantes. Se puede reproducir en un entorno de campo con materiales de bajo costo. La estandarización de estas condiciones es el requisito mínimo para garantizar resultados consistentes entre sesiones de clasificación. Esta limitación es documentada como una restricción del sistema en la sección de conclusiones.

3.5.2 Velocidad de procesamiento

El sistema actual clasifica una manzana a la vez, con un tiempo de procesamiento de aproximadamente 2 a 5 segundos por imagen en hardware convencional con GPU. Para un pequeño productor que clasifica su producción manualmente, el sistema ofrece una guía objetiva que reduce la variabilidad del criterio humano sin requerir velocidades industriales. No es adecuado para líneas de empaque de alta velocidad.

3.5.3 Facilidad de uso

La implementación actual requiere conocimientos básicos de Python y entorno Jupyter Notebook. Esta restricción limita el perfil de usuario al técnico o profesional con formación en informática. El desarrollo de una interfaz gráfica simple, identificado como primera línea de trabajo futuro, permitiría extender el uso a operarios sin formación técnica, operando desde una tablet o computadora de escritorio.

3.5.4 Variabilidad entre variedades

El modelo fue entrenado y evaluado exclusivamente con manzanas de una variedad particular capturadas en condiciones controladas. La aplicación a otras variedades (Granny Smith, Fuji, Gala) requeriría la recolección de nuevas muestras etiquetadas y posiblemente el reentrenamiento del modelo. Esta limitación es consistente con el alcance prototípico del trabajo.

4. Diseño del Sistema

Esta sección describe las decisiones de diseño que dan forma al prototipo, la arquitectura general del sistema, el protocolo de captura de imágenes, la construcción y organización del dataset, y el modelo de datos utilizado. Estas decisiones fueron adoptadas con criterio de viabilidad práctica para un pequeño productor, priorizando el bajo costo y la reproducibilidad sobre la escala industrial.

4.1 Arquitectura general del prototipo

El prototipo sigue una arquitectura de tres etapas secuenciales: captura, procesamiento y clasificación.

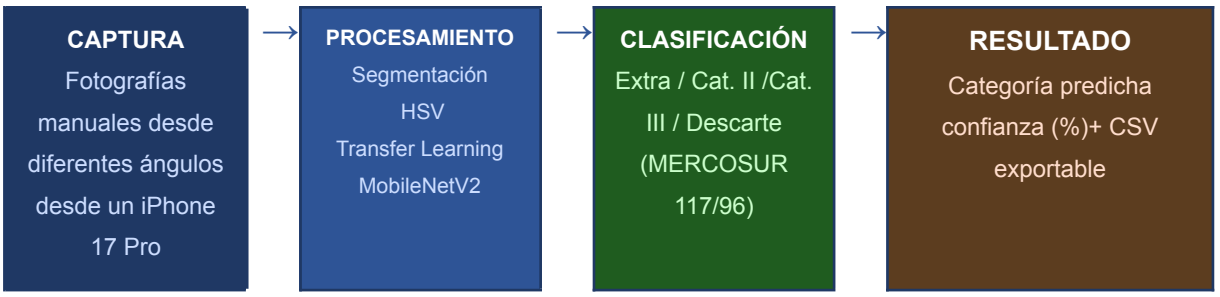


Figura 4.1. Flujo general del sistema

La etapa de captura es realizada por el operario de forma manual, siguiendo el protocolo definido en la sección 4.2. La etapa de procesamiento implementa dos métodos alternativos (HSV y Transfer Learning) cuyos detalles se desarrollan en la Sección 5. La etapa de clasificación combina las predicciones de todas las fotos de una misma manzana promediando las probabilidades individuales, y emite una categoría final junto con el nivel de confianza.

El sistema opera en modo fuera de línea (offline): las fotografías se toman primero y luego se procesan en lote. No requiere conexión a internet ni infraestructura de servidor. Todo el procesamiento ocurre localmente en la computadora del usuario.

4.2 Protocolo de captura de imágenes

El protocolo de captura define las condiciones mínimas que deben respetarse para garantizar resultados consistentes entre sesiones de clasificación. Su diseño busca el equilibrio entre reproducibilidad y accesibilidad. Todas las condiciones requeridas pueden cumplirse con materiales disponibles en cualquier entorno doméstico o de pequeña producción, sin equipamiento especializado.

4.2.1 Equipamiento utilizado

Componente	Especificación utilizada	Requisito mínimo recomendado
Cámara	iPhone 17 Pro (cámara principal trasera, 48 MP)	Cualquier smartphone moderno o webcam ≥ 8 MP
Fondo	Bandeja de cartón corrugado negro (22 × 30 cm aprox.)	Superficie plana, color oscuro y uniforme, sin texturas brillantes
Iluminación	Luz natural difusa (día nublado, interior)	Luz uniforme sin sombras duras; evitar luz solar directa
Posición de la cámara	Cenital (perpendicular a la bandeja), sin zoom	Cenital o ligeramente inclinada; distancia fija entre tomas
Encuadre	La manzana ocupa aproximadamente el 60–70% del cuadro	La fruta debe estar centrada y completamente visible

Tabla 4.1. Equipamiento utilizado y requisitos mínimos recomendados para la captura estandarizada de imágenes de manzanas.

4.2.2 Procedimiento de captura

Para cada manzana se obtiene un conjunto de fotografías desde distintos ángulos siguiendo los pasos indicados a continuación:

- Colocar la bandeja de fondo en una superficie plana, bajo condiciones de iluminación difusa y uniforme.
- Posicionar la manzana en el centro de la bandeja con el cáliz hacia arriba (ángulo 0°).
- Fotografiar desde posición cenital, manteniendo la cámara perpendicular a la bandeja y a distancia constante.
- Rotar la manzana 90° sobre su eje vertical y repetir la fotografía. Repetir el proceso hasta completar los cuatro ángulos: 0°, 90°, 180° y 270°.

- Opcionalmente, tomar fotografías adicionales de ángulos específicos donde se observen defectos, para incrementar la cobertura visual de la superficie del fruto.

La rotación se realiza sobre el eje vertical de la fruta, desde el pedúnculo hasta el cáliz, y no sobre el eje ecuatorial. Esta decisión permite que los cuatro ángulos obtenidos cubran la totalidad de la superficie lateral del fruto, que es donde suelen manifestarse los principales defectos superficiales contemplados por la normativa MERCOSUR.

4.2.3 Justificación del diseño del protocolo

La elección de cuatro ángulos equidistantes responde a un criterio de cobertura mínima suficiente: cuatro vistas a 90° entre sí permiten observar la totalidad de la superficie lateral de la manzana sin superposición significativa. Este enfoque es consistente con la literatura en clasificación de frutas por múltiples vistas, donde se ha demostrado que el uso de más de una imagen por fruto mejora la precisión de clasificación respecto al análisis de una única imagen.

La elección de fondo oscuro (bandeja negra) facilita la segmentación de la fruta por contraste de color, aspecto relevante para el Método HSV desarrollado en la Sección 5.1. La iluminación difusa, preferiblemente día nublado o luz artificial indirecta, reduce los reflejos especulares que dificultan la detección de defectos superficiales en frutos de epidermis brillante como las manzanas rojas.

La resolución de las imágenes del iPhone 17 Pro (48 MP en el sensor principal) supera ampliamente los requisitos del sistema. Para el procesamiento, las imágenes son redimensionadas a 224 × 224 píxeles (requerimiento de MobileNetV2) o a 800 píxeles en el lado mayor (método HSV), reduciendo el tiempo de procesamiento sin pérdida significativa de información relevante para la clasificación.

4.3 Construcción del dataset

4.3.1 Origen y etiquetado de las muestras

El dataset fue construido con manzanas de la variedad Red Delicious, la variedad de mayor superficie cultivada en el Alto Valle del Río Negro (aproximadamente el 60% del total según SENASA, 2021). Las manzanas fueron obtenidas de la producción de Kleppe S.A. y clasificadas previamente por el sistema de clasificación óptica Spectrim en operación en sus líneas de empaque. La categoría asignada por el sistema industrial fue verificada y confirmada a ojo por un operario con experiencia en clasificación de la misma empresa, garantizando la veracidad de las etiquetas del dataset.

Esta metodología de etiquetado, clasificación industrial confirmada por experto humano, garantiza que las etiquetas del dataset son confiables y objetivas, eliminando el sesgo de la clasificación visual manual sin respaldo técnico. Es una fortaleza metodológica relevante para la validez de los resultados obtenidos.

4.3.2 Composición del dataset

Categoría MERCOSUR	Carpeta	Manzanas	Total fotos	Train (80%)	Val (20%)
Extra	Categoría 1	4	28	23	5
Categoría II	Categoría 2	4	29	24	5
Categoría III	Categoría 3	4	26	21	5
Descarte	Categoría 4	4	28	23	5
TOTAL	**4 cat.**	**16**	**111**	**91**	**20**

Tabla 4.2. Composición del dataset por categoría MERCOSUR.

Cada manzana fue fotografiada desde múltiples ángulos, obteniéndose entre 6 y 8 fotografías por fruto dependiendo de las características visuales observadas. Las fotografías adicionales a los cuatro ángulos base fueron tomadas cuando se identificaron defectos en zonas específicas que requerían mayor cobertura.

4.3.3 Organización de archivos

Las imágenes se organizaron en una estructura de directorios jerárquica que refleja las categorías MERCOSUR, compatible con las herramientas de carga automática de datasets de las librerías PyTorch (ImageFolder) y OpenCV utilizadas en el desarrollo:

```
dataset/
├── Categoria 1/        ← Categoría Extra-MERCOSUR
│   ├── Manzana 1/
│   │   ├── IMG_2419.JPG
│   │   ├── IMG_2420.JPG
│   │   └── ... (6–8 fotos)
```

```
│   ├── Manzana 2/
│   └── ...
├── Categoria 2/        ← Categoría II MERCOSUR
├── Categoria 3/        ← Categoría III  MERCOSUR
└── Categoria 4/        ← Descarte MERCOSUR
```

Figura 4.2. - Estructura de directorios del dataset.

4.3.4 División train/validación

El dataset fue dividido en conjunto de entrenamiento (80%) y conjunto de validación (20%) mediante selección aleatoria con semilla fija (seed = 42), garantizando la reproducibilidad de la partición. La semilla fija asegura que la misma división se obtenga en cada ejecución del código, permitiendo comparaciones consistentes entre experimentos.

Las imágenes de validación no fueron vistas por el modelo durante el entrenamiento en ningún momento del proceso. Esta separación estricta es la que permite que las métricas reportadas (precisión, recall, F1-score) sean una estimación válida del desempeño del sistema ante datos nuevos, y no una medición del ajuste sobre datos conocidos. No obstante, debe considerarse que, al contar con 20 imágenes de validación en total (5 por categoría), cada predicción incorrecta equivale a 5 puntos porcentuales de diferencia en la métrica de accuracy. Esto introduce variabilidad en los resultados, que debe tenerse en cuenta al interpretar las métricas reportadas en la Sección 6.

4.4 Modelo de datos

El sistema maneja dos tipos principales de datos a lo largo del proceso de clasificación:

Entidad	Tipo	Descripción
Imagen	Archivo .JPG	Fotografía individual de una manzana desde un ángulo determinado. Atributos: nombre de

		archivo, categoría (carpeta padre), manzana de origen (subcarpeta).
Manzana	Conjunto de imágenes	Conjunto de todas las fotografías de un mismo fruto. Atributos: carpeta, categoría real, n° de fotos, categoría predicha, confianza, porcentaje de defectos por tipo.
Resultado de clasificación	Registro CSV	Resultado final por manzana. Campos: manzana, categoría real, categoría predicha, correcto (bool), confianza (%), probabilidades por clase.
Modelo entrenado	Archivo .pth (PyTorch)	Pesos del modelo MobileNetV2 con fine-tuning. Se guarda automáticamente la mejor versión según val_accuracy durante el entrenamiento.

Tabla 4.3. Modelo de datos del sistema.

Los resultados de cada sesión de clasificación se exportan automáticamente a un archivo CSV que incluye, para cada imagen evaluada, la categoría real, la categoría predicha, si la predicción fue correcta, el nivel de confianza y las probabilidades asignadas a cada una de las cuatro categorías. Este archivo permite el análisis posterior de los resultados y su incorporación a registros de calidad del productor. Las decisiones de diseño documentadas en esta sección, en particular el protocolo de captura, la estructura del dataset y el modelo de datos, constituyen los insumos directos del desarrollo descrito en la Sección 5. Cualquier replicación del experimento deberá respetar estas especificaciones para obtener resultados comparables.

5. Desarrollo del Prototipo

El presente capítulo describe el desarrollo del prototipo implementado para la clasificación automática de manzanas según su calidad comercial. Se documentan los dos métodos desarrollados: un enfoque basado en segmentación por color en el espacio HSV y un enfoque basado en aprendizaje profundo mediante Transfer Learning con MobileNetV2.

Para cada método se presenta su fundamento técnico, el proceso de implementación, los parámetros utilizados, las etapas de calibración o entrenamiento y las principales dificultades

encontradas durante el desarrollo. Los resultados cuantitativos obtenidos por ambos métodos se presentan posteriormente en la Sección 6.

5.1 Método 1 — Segmentación por color HSV

5.1.1 Fundamento técnico

El espacio de color HSV (Hue, Saturation, Value) es una representación cromática que separa la información visual de una imagen en tres componentes: tono, saturación y valor. El tono (Hue) representa el color dominante del píxel; la saturación (Saturation) expresa la pureza o intensidad del color; y el valor (Value) indica su luminosidad.

A diferencia del espacio RGB, donde cada canal combina información de color e iluminación, el espacio HSV permite definir rangos cromáticos de forma más intuitiva. Esta característica lo vuelve adecuado para tareas de segmentación por color, especialmente cuando se busca identificar regiones visualmente diferenciables dentro de una imagen.

En el presente trabajo, el método HSV se utilizó para detectar defectos superficiales en manzanas. El proceso se organizó en dos etapas principales: primero, la segmentación de la fruta respecto del fondo; luego, la detección de regiones internas de la fruta cuyos valores HSV coincidieran con los rangos asociados a defectos visibles.

5.1.2 Implementación

La implementación se realizó en Python 3.12 utilizando la librería OpenCV 4.x. El proceso completo para cada imagen consta de los siguientes pasos:

- Carga y redimensionado: la imagen se carga con cv2.imread() y se redimensiona a un máximo de 800 píxeles en su lado mayor para acelerar el procesamiento, manteniendo la proporción original. Esto permite reducir el costo computacional sin alterar significativamente la información visual relevante.
- Conversión al espacio HSV: La imagen se convierte desde el espacio BGR utilizado por OpenCV al espacio HSV mediante cv2.cvtColor(img, cv2.COLOR_BGR2HSV).
- Segmentación del fruto: se aplican máscaras de color para los rangos HSV correspondientes a los colores naturales de la manzana (rojo, naranja, amarillo, verde). El fondo oscuro de la bandeja queda excluido por tener saturación y valor muy bajos. Se aplican operaciones morfológicas (cierre y apertura) para limpiar la máscara y eliminar ruido. Finalmente, se conserva únicamente el contorno de mayor área, que corresponde a la fruta.

- Detección de defectos: sobre la máscara de la fruta, se aplican rangos HSV específicos para cada tipo de defecto. Se calculan las operaciones morfológicas de apertura para eliminar detecciones espurias menores a 5×5 píxeles. Se computa el porcentaje de superficie afectada como la relación entre los píxeles detectados como defecto y el total de píxeles de la fruta.
- Clasificación: se suman los porcentajes de defectos graves (machucón, quemado de sol, herida, sarna) y se comparan con los umbrales definidos por la normativa MERCOSUR para asignar la categoría final.

Los rangos HSV finales calibrados para cada tipo de defecto y para la segmentación de la fruta fueron los siguientes:

Región/Defecto	H min	H max	S min / V min	S max / V max
Fruta — rojo (rango 1)	0	10	100 / 80	255 / 255
Fruta — rojo (rango 2)	165	180	100 / 80	255 / 255
Fruta — naranja	10	25	100 / 80	255 / 255
Fruta — amarillo	20	32	80 / 120	255 / 255
Machucón	5	12	120 / 20	255 / 80
Quemado de sol	0	180	0 / 15	50 / 70
Herida abierta	35	80	40 / 15	150 / 90
Sarna	20	50	40 / 15	130 / 80

Tabla 5.1. Rangos HSV finales calibrados para segmentación y detección de defectos.

Los umbrales de clasificación final, basados en la Tabla II de la normativa MERCOSUR GMC RES. 117/1996, fueron:

Categoría	Máx. defectos graves (%)	Máx. defectos totales (%)	Resultado si se supera

Extra	2%	5%	Categoría I
Categoría I	6%	12%	Categoría II
Categoría II	10%	20%	Descarte

Tabla 5.2. Umbrales de clasificación basados en MERCOSUR GMC RES. 117/1996, Tabla II.

5.1.3 Proceso de calibración

Los rangos HSV no pueden determinarse de forma completamente teórica, ya que dependen de las condiciones concretas de captura, iluminación, cámara utilizada, coloración de las frutas y características visuales del fondo. Por este motivo, la calibración se realizó mediante un proceso iterativo basado en inspección visual y ajuste progresivo de rangos.

En primer lugar, se implementó una herramienta de diagnóstico que divide cada imagen en una grilla de 6×6 celdas y calcula los valores promedio de H, S y V para cada región. Esta visualización permite identificar los valores cromáticos aproximados correspondientes a zonas sanas, defectos visibles, fondo y regiones conflictivas como el cáliz.

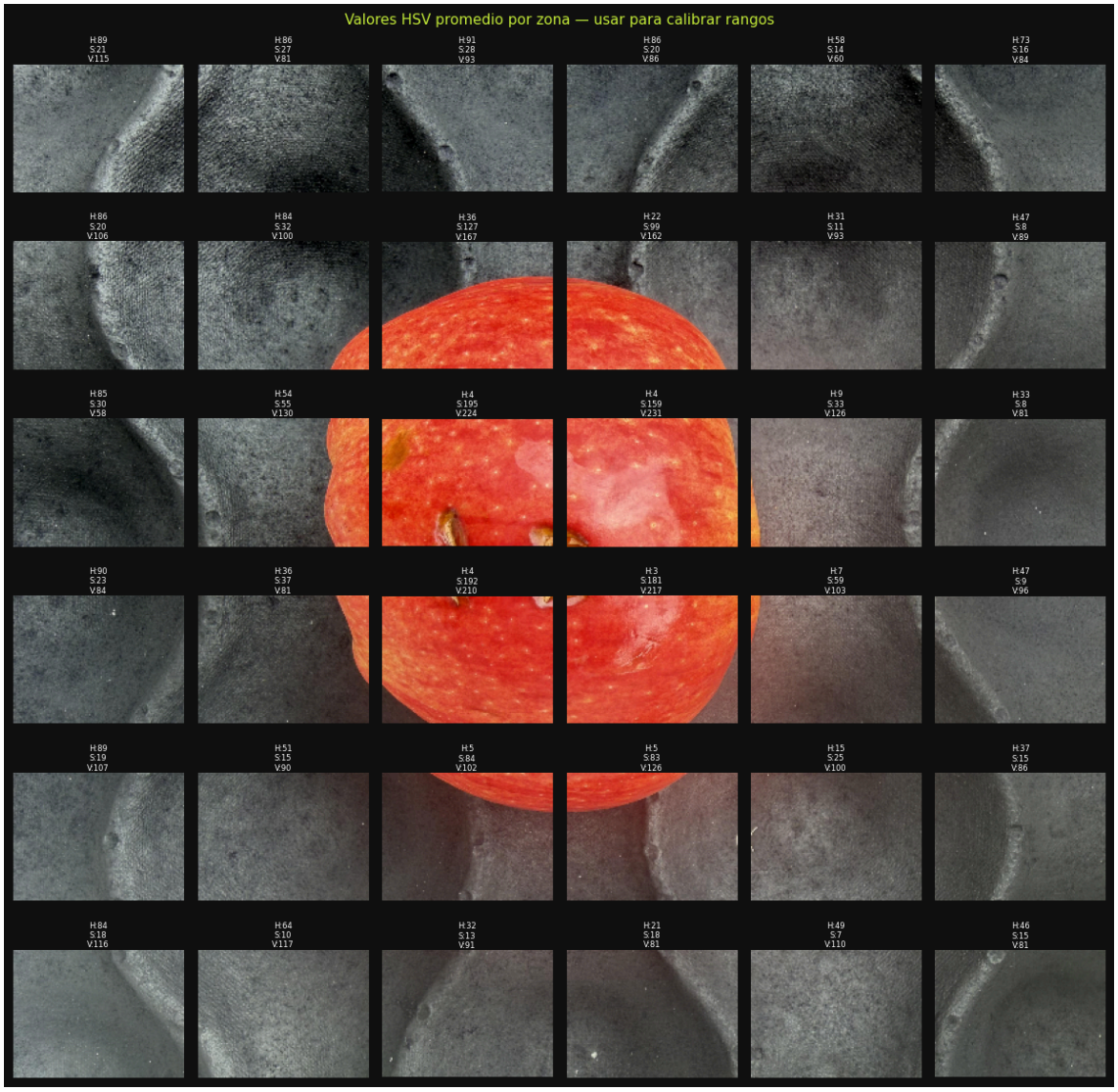


Figura 5.1. Grilla de diagnóstico HSV utilizada para la calibración de rangos cromáticos

La imagen se divide en una grilla de 6×6 celdas y se muestran los valores promedio de tono, saturación y valor de cada zona.

A partir de esta información se definieron rangos iniciales para cada tipo de defecto. Luego, dichos rangos se aplicaron sobre imágenes representativas del dataset y se evaluó visualmente la correspondencia entre la máscara generada y las zonas defectuosas reales. Posteriormente, los límites inferiores y superiores de cada rango fueron ajustados para reducir falsos positivos, es decir, regiones sanas detectadas como defectuosas, y falsos negativos, es decir, defectos reales no detectados.

El proceso de calibración se repitió sobre imágenes de distintas categorías para verificar que los rangos no funcionarán únicamente en un caso particular, sino que fueran aplicables a una variedad razonable de manzanas dentro del dataset.

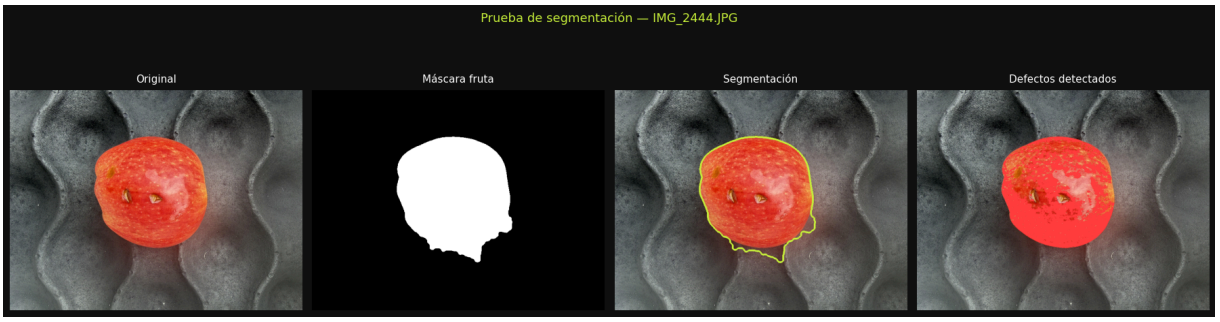


Figura 5.2. Ejemplo del proceso de segmentación HSV.

Se muestra la imagen original, la máscara binaria de la fruta, el contorno de segmentación sobre la imagen original y las regiones detectadas como defectos superficiales.

5.1.4 Dificultades encontradas

Durante el desarrollo y evaluación del método HSV se identificaron varias limitaciones técnicas.

En primer lugar, se observó contaminación por el fondo. La bandeja de cartón negro presentaba reflejos, texturas y variaciones de tono que en algunos casos eran captadas por las máscaras de segmentación, generando falsos positivos fuera de la fruta. Esta dificultad se mitigó aumentando los valores mínimos de saturación utilizados para detectar la fruta, especialmente en los rangos rojos y naranjas.

Otra dificultad relevante fue la zona del cáliz. Esta región presenta naturalmente una coloración amarillo-naranja o marrón que puede confundirse con defectos superficiales. Para reducir este problema fue necesario ajustar los rangos asociados al machucón y limitar los valores de luminosidad, de modo de capturar preferentemente zonas más oscuras.

También se detectó una limitación estructural del método: ciertas cicatrices o heridas cicatrizadas presentan valores HSV muy similares a los de la epidermis sana de la manzana. En estos casos, al no existir una diferencia cromática suficientemente marcada, el método basado únicamente en color no logra distinguir de forma confiable entre superficie sana y superficie defectuosa.

Finalmente, el método mostró sensibilidad a las condiciones de iluminación. Cambios en la intensidad o temperatura de color de la luz modifican los valores HSV registrados por la cámara, pudiendo desplazar los defectos fuera de los rangos previamente calibrados. Esta limitación evidencia la dependencia del método respecto de condiciones de captura relativamente controladas.

Estas dificultades motivaron la incorporación de un segundo enfoque basado en aprendizaje profundo, capaz de aprovechar no solo información cromática, sino también patrones de textura, forma y contexto visual.

5.2 Método 2 — Transfer Learning con MobileNetV2

5.2.1 Fundamento técnico

Las redes neuronales convolucionales, o CNN (Convolutional Neural Networks), son arquitecturas de aprendizaje profundo diseñadas especialmente para el procesamiento de imágenes. Su funcionamiento se basa en capas convolucionales que aplican filtros aprensibles sobre la imagen de entrada para detectar patrones locales como bordes, texturas, gradientes y combinaciones espaciales de color.

A medida que la información avanza por la red, las primeras capas tienden a aprender características visuales simples, mientras que las capas más profundas aprenden representaciones más complejas y específicas. Esto permite que una CNN extraiga características relevantes de una imagen sin necesidad de definir manualmente todos los criterios visuales de clasificación.

MobileNetV2 [8] es una arquitectura CNN diseñada para aplicaciones con restricciones de cómputo. Su diseño se basa en bloques de botella invertidos y convoluciones separables en profundidad, que reducen significativamente la cantidad de parámetros y operaciones requeridas en comparación con arquitecturas más pesadas, manteniendo una capacidad predictiva competitiva.

El Transfer Learning Pan y Yang [7] permite aprovechar un modelo previamente entrenado en una tarea general, como la clasificación de imágenes de ImageNet, y adaptarlo a una tarea específica. En este caso, se utilizó MobileNetV2 pre entrenada como extractor de características visuales, reemplazando su clasificador original por uno nuevo orientado a las cuatro categorías de calidad consideradas en este trabajo: Extra, Categoría II, Categoría III y Descarte.

5.2.2 Arquitectura del modelo

La arquitectura final combina la base convolucional de MobileNetV2 con un clasificador personalizado. MobileNetV2 recibe imágenes de entrada de 224×224×3 píxeles y produce un mapa de características que resume información visual relevante de la imagen. Sobre esta base se

agregaron capas adicionales para adaptar el modelo al problema específico de clasificación de manzanas.

Componente	Parámetros	Descripción
MobileNetV2 base	2.258.604 (congelados)	Extractor de características pre entrenado en ImageNet. Recibe una imagen de 224×224×3 píxeles. Al estar “congelado”, sus pesos no se modifican en la Fase 1. Conserva todo lo que aprendió sin alterar el conocimiento.
GlobalAveragePooling2D	0	Reduce el mapa de características a un vector de 1280 dimensiones.
Dropout (p=0.3)	0	Desactiva aleatoriamente el 30% de las neuronas durante el entrenamiento para reducir el sobreajuste. Obliga al modelo a no depender de ninguna neurona específica, distribuyendo el conocimiento entre todas.
Dense (128 neuronas, ReLU)	163.968	Es la capa de clasificación intermedia. Las 128 neuronas reciben el vector de 1280 características y aprenden a combinarlas para distinguir entre las cuatro categorías de manzanas. La función de ReLU (Rectified Linear Unit) convierte los valores negativos en 0 y deja pasar los positivos sin cambio. Esto hace que el modelo sea no-linealidad, haciendo que aprenda patrones complejos.
Dropout (p=0.2)	0	Segunda capa de regularización. Evita que el modelo dependa demasiado de combinaciones específicas de las 128 características aprendidas.
Dense (4 neuronas, Softmax)	516	Capa de salida con cuatro neuronas, una por categoría: Extra, Categoría II, Categoría III y

		Descarte. Softmax convierte las salidas en probabilidades que suman 1. La categoría con mayor probabilidad es la predicción final.
TOTAL ENTRENABLES	**164.484**	**Corresponde al clasificador agregado sobre MobileNetV2. En la Fase 2 se suman las capas descongeladas**

Tabla 5.3. - Arquitectura del modelo MobileNetV2 con clasificador personalizado.

La decisión de congelar inicialmente la base convolucional responde al tamaño reducido del dataset. Entrenar todos los parámetros del modelo desde el inicio aumentaría considerablemente el riesgo de sobreajuste. Por este motivo, en una primera etapa se entrenó solamente el nuevo clasificador. Luego, en una segunda etapa, se descongelaron algunas capas finales de MobileNetV2 para ajustar parcialmente el extractor de características al dominio específico de las imágenes de manzanas.

5.2.3 Data augmentation

Dado el tamaño reducido del conjunto de entrenamiento, se aplicó data augmentation para aumentar artificialmente la variabilidad de las imágenes disponibles. Esta técnica consiste en generar transformaciones aleatorias sobre las imágenes durante el entrenamiento, de modo que el modelo observe distintas versiones de una misma fotografía en diferentes épocas.

Las transformaciones utilizadas fueron las siguientes:

Transformación	Parámetro	Justificación
Rotación aleatoria	±30°	Simula variaciones en el ángulo de captura manual de la fotografía.
Espejo horizontal	p = 0.5	Genera una imagen especular. La fruta tiene simetría rotacional, lo que hace válida esta transformación.
Espejo vertical	p = 0.5	Simula fotografías tomadas con la fruta invertida.

Variación de brillo y contraste	±30%	Simula diferentes condiciones de iluminación.
Zoom aleatorio	80%–100%	Simula variaciones en la distancia entre la cámara y la fruta.

Tabla 5.4. Transformación de data augmentation aplicadas durante el entrenamiento.

Estas transformaciones buscan mejorar la capacidad de generalización del modelo, reduciendo la dependencia respecto de una orientación, escala o condición lumínica particular.

5.2.4 Proceso de entrenamiento

El entrenamiento se realizó en dos fases consecutivas. Antes de describir cada fase, se resumen los principales parámetros utilizados.

Parámetro	Valor	Qué significa
Optimizador	Adam	Adam (Adaptive Moment Estimation) es el algoritmo que ajusta los pesos del modelo después de cada lote de imágenes. A diferencia del descenso de gradiente clásico que usa la misma tasa de aprendizaje para todos los parámetros, Adam ajusta la tasa individualmente para cada parámetro según su historial de gradientes. Es el optimizador más utilizado en deep learning por su convergencia rápida y estabilidad.
Función de pérdida	CrossEntropyLoss	Mide qué tan equivocadas están las predicciones del modelo. Para un problema de 4 clases, calcula cuánto difieren las probabilidades predichas de las probabilidades reales. Un valor de pérdida bajo indica predicciones cercanas a la realidad. El modelo ajusta sus pesos para minimizar esta pérdida en cada iteración.
Learning rate (Fase 1)	0.001	La tasa de aprendizaje controla el tamaño del paso con que el optimizador ajusta los pesos. Un valor de 0.001 es moderado, suficientemente grande para aprender rápido, suficientemente pequeño para no sobrepasar el mínimo

		óptimo. Se aplica al clasificador nuevo (164.484 parámetros) en la Fase 1.
Learning rate (Fase 2)	0.00005 / 0.0001	En la Fase 2 se usan dos tasas diferentes: 0.00005 (cincuenta veces menor) para las últimas 3 capas de MobileNetV2 descongeladas, y 0.0001 para el clasificador. La tasa muy baja en MobileNetV2 garantiza ajustes mínimos que preservan el conocimiento pre entrenado sin borrarlo.
Scheduler	StepLR (step=7, γ=0.5)	El scheduler reduce la tasa de aprendizaje automáticamente cada 7 épocas multiplicándose por 0.5. Esto permite un aprendizaje más rápido al principio, más fino y preciso en las épocas finales, evitando oscilaciones alrededor del mínimo óptimo.
Batch size	8	El modelo procesa 8 imágenes por vez antes de actualizar sus pesos. Un batch pequeño (8) es adecuado para datasets reducidos, permite actualizaciones más frecuentes y agrega cierta variabilidad al aprendizaje, actuando como regularizador natural.
Early Stopping	paciencia = 8	Si la precisión de validación no mejora durante 8 épocas consecutivas, el entrenamiento se detiene automáticamente y se restaura el mejor modelo encontrado hasta ese momento. Esto evita el sobreajuste (que el modelo memorice las fotos de entrenamiento) y el desperdicio de tiempo computacional en épocas sin mejora.
WeightedRandomSampler	Pesos inversamente proporcionales	Como las categorías no tienen exactamente la misma cantidad de imágenes, se usa un muestreador ponderado que da más probabilidad de ser seleccionadas a las imágenes de clases con menos ejemplos. Esto evita que el modelo se sesgue hacia las clases más representadas.

Tabla 5.5. Parámetros principales del entrenamiento

3.2.4.1 Fase 1 - Entrenamiento del clasificador

Durante la primera fase, las capas de la base MobileNetV2 se mantuvieron congeladas. Solo se entrenaron los parámetros del clasificador agregado, equivalente a 164484 parámetros entrenables.

Se utilizó el optimizador Adam con una tasa de aprendizaje inicial de 0.001. El entrenamiento se detuvo mediante Early Stopping en la época 12, luego de no observarse una mejora sostenida en la precisión de validación. El mejor modelo de esta fase alcanzó un 70% de precisión de validación en la época 4.

```
Fase 1

Época 01/30 | Train: 30.8% | Val: 30.0%
Época 02/30 | Train: 44.0% | Val: 25.0%
Época 03/30 | Train: 38.5% | Val: 50.0%
Época 04/30 | Train: 52.7% | Val: 70.0%
Época 05/30 | Train: 60.4% | Val: 50.0%
Época 06/30 | Train: 48.4% | Val: 45.0%
Época 07/30 | Train: 51.6% | Val: 40.0%
Época 08/30 | Train: 45.1% | Val: 40.0%
Época 09/30 | Train: 54.9% | Val: 55.0%
Época 10/30 | Train: 59.3% | Val: 60.0%
Época 11/30 | Train: 61.5% | Val: 70.0%
Época 12/30 | Train: 67.0% | Val: 50.0%
```

Figura 5.3. Registro de entrenamiento de la Fase 1.

Se muestran los valores de precisión obtenidos en entrenamiento (Train) y validación (Val) para cada época durante el entrenamiento inicial del clasificador, manteniendo congelada de la base convolucional MobileNetV2

3.2.4.2 Fase 2 - Fine-tuning

En la segunda fase se cargó el mejor modelo obtenido en la Fase 1 y se descongelaron las últimas tres capas de MobileNetV2. El objetivo de esta etapa fue ajustar parcialmente el extractor de características al dominio específico de las imágenes de manzanas, sin modificar de forma abrupta el conocimiento previamente aprendido en ImageNet.

Para ello, se utilizaron tasas de aprendizaje diferenciadas: una tasa menor para las capas descongeladas de MobileNetV2 y una tasa levemente superior para el clasificador. El mejor modelo final alcanzó un 80% de precisión de validación en la época 3 de esta segunda fase.

```
Fase 2

Época 01/20 | Train: 59.3% | Val: 70.0%
Época 02/20 | Train: 56.0% | Val: 70.0%
Época 03/20 | Train: 63.7% | Val: 80.0%
Época 04/20 | Train: 62.6% | Val: 80.0%
Época 05/20 | Train: 76.9% | Val: 75.0%
Época 06/20 | Train: 65.9% | Val: 65.0%
Época 07/20 | Train: 64.8% | Val: 75.0%
Época 08/20 | Train: 59.3% | Val: 75.0%
Época 09/20 | Train: 72.5% | Val: 70.0%
Época 10/20 | Train: 71.4% | Val: 75.0%
Época 11/20 | Train: 73.6% | Val: 70.0%
```

Figura 5.4. Registro de entrenamiento de la Fase 2.

Se muestran los valores de precisión obtenidos en entrenamiento (Train) y validación (Val) durante la etapa de fine-tuning, en la cual se descongelaron las últimas capas de MobileNetV2 para ajustar parcialmente el modelo al dominio específico de imágenes de manzanas.

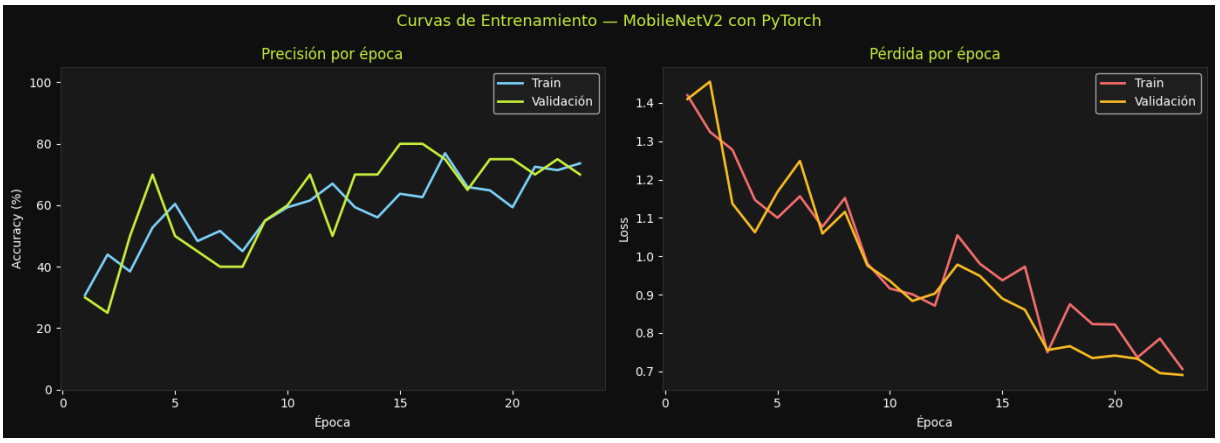


Figura 5.5. Curvas de entrenamiento del modelo MobileNetV2.

Se muestra la evolución de la precisión y la pérdida durante el entrenamiento del modelo MobileNetV2. En el gráfico de precisión se observa una tendencia general ascendente tanto en entrenamiento como en validación, alcanzando un valor máximo de validación del 80%. Las

fluctuaciones presentes en la curva de validación son esperables debido al tamaño reducido del conjunto de validación, donde cada imagen representa una variación porcentual considerable.

En el gráfico de pérdida se observa una disminución progresiva desde valores cercanos a 1,4 hasta aproximadamente 0,7, lo que indica que el modelo fue reduciendo el error de clasificación a lo largo de las épocas. No se evidencia un sobreajuste severo, ya que las curvas de entrenamiento y validación no presentan una divergencia marcada. En conjunto, ambos gráficos indican que el modelo logró aprender patrones relevantes del dataset y mejorar su capacidad de generalización durante el proceso de entrenamiento.

5.2.5 Clasificación de manzanas individuales

El sistema final opera a nivel de manzana y no únicamente a nivel de imagen individual. Para cada fruta se procesan todas las fotografías disponibles, tomadas desde diferentes ángulos. El modelo genera para cada imagen un vector de probabilidades correspondiente a las cuatro categorías posibles.

Luego, las probabilidades obtenidas para todas las fotografías de una misma manzana se promedian. La categoría final asignada corresponde a aquella con mayor probabilidad promedio.

Este enfoque mejora la robustez de la clasificación, ya que permite integrar información visual proveniente de múltiples perspectivas. Una fotografía donde el defecto no es visible puede ser compensada por otra imagen de la misma fruta donde sí se observa la zona afectada.

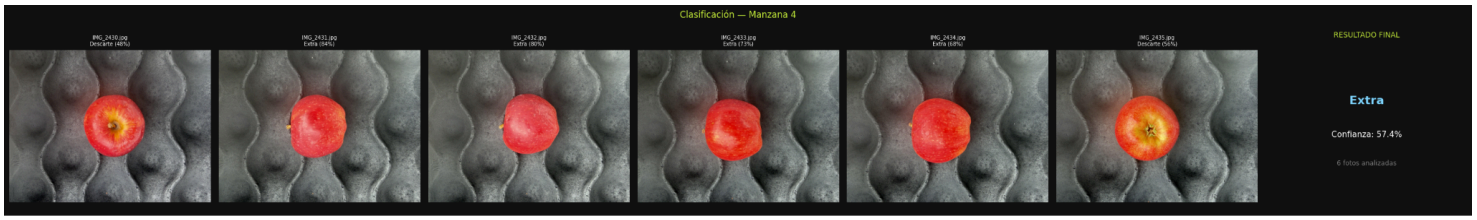


Figura 5.6. Ejemplo de clasificación de una manzana mediante múltiples fotografías.

Se muestran las predicciones individuales por imagen y el resultado final obtenido a partir del promedio de probabilidades.

5.3 Herramientas y tecnologías utilizadas

La siguiente tabla resume las principales herramientas utilizadas durante el desarrollo del prototipo:

Herramienta	Versión	Uso en el proyecto

Python	3.12.2	Lenguaje principal de programación.
PyTorch	2.5.1+cu121	Framework de deep learning. Implementación del modelo, entrenamiento y evaluación.
torchvision	0.20.1	MobileNetV2 preentrenada, data augmentation e ImageFolder para carga del dataset.
OpenCV	4.x	Procesamiento de imágenes, conversión a HSV, operaciones morfológicas y segmentación.
NumPy	2.x	Operaciones matriciales y cálculo de porcentajes de superficie afectada.
scikit-learn	1.x	Generación del reporte de clasificación (precision, recall, F1-score) y matriz de confusión.
Matplotlib / Seaborn	3.x	Visualización de curvas de entrenamiento, matrices de confusión y gráficos de barras.
Pandas	2.x	Exportación de resultados a CSV y análisis de los datos de validación.
Jupyter Notebook (VS Code)	—	Entorno de desarrollo, experimentación iterativa y documentación del proceso.
NVIDIA CUDA	12.1	Aceleración por GPU en el entrenamiento del modelo.
GPU: NVIDIA RTX 3080	10 GB VRAM	Hardware de entrenamiento. Redujo el tiempo de entrenamiento significativamente respecto a CPU.

Tabla 5.6. Stack tecnológico utilizado en el desarrollo del prototipo.

6. Resultados y Evaluación

Este capítulo presenta los resultados obtenidos por los dos métodos de clasificación desarrollados, su comparación directa y la evaluación de la hipótesis de trabajo planteada en la Sección 1.4.

En el caso del método basado en segmentación HSV, la evaluación se realizó a nivel de manzana sobre las 16 frutas del dataset completo, considerando entre 6 y 8 fotografías por fruto. En cambio, para el método basado en Transfer Learning, los resultados se reportan sobre el conjunto de validación de 20 imágenes (5 por categoría), que no fue utilizado durante el entrenamiento.

Esta diferencia metodológica debe tenerse en cuenta al interpretar la comparación entre ambos enfoques.

6.1 Resultados del Método HSV

6.1.1 Resultados por manzana

El método HSV fue evaluado sobre las 16 manzanas del dataset completo, distribuidas en cuatro categorías (4 manzanas por categoría), con entre 6 y 8 fotografías por fruto. La clasificación final de cada manzana se determinó tomando el peor caso observado entre todas sus imágenes, con el objetivo de reflejar una evaluación conservadora de la calidad. Los resultados obtenidos se presentan en la Tabla 6.1.

Categoría real	Correctas	Total	Precisión
Extra	0	4	0%
Categoría II	1	4	25%
Categoría III	0	4	0%
Descarte	0	4	0%
TOTAL	1	16	6,2%

Tabla 6.1. Resultados del Método HSV por categoría MERCOSUR.

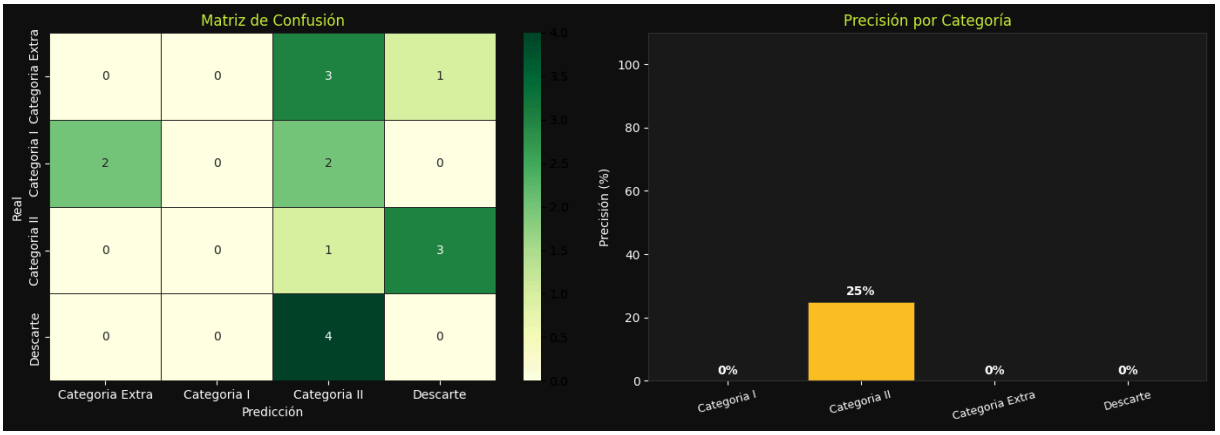


Figura 6.1. Resultados del Método HSV: matriz de confusión (izquierda) y precisión por categoría (derecha).

6.1.2 Análisis de los resultados HSV

La precisión general obtenida por el método HSV fue de 6,2% (1 manzana correctamente clasificada sobre 16), lo que indica un desempeño insuficiente para el problema abordado. El análisis de los errores muestra un patrón sistemático: la mayoría de las manzanas fueron clasificadas como Categoría II, independientemente de su clase real.

Este comportamiento se explica porque los porcentajes de defecto estimados por el sistema tendieron a concentrarse en el rango correspondiente a esa categoría. En particular, los falsos positivos generados por reflejos del fondo y por la coloración natural del cáliz incrementan artificialmente la superficie afectada, mientras que varios defectos reales no pudieron ser detectados debido a su similitud cromática con la epidermis sana.

Las causas específicas de cada error se detallan a continuación:

- Extra (0%): el sistema generó falsos positivos en todas las manzanas sin defecto, interpretando reflejos de la bandeja y la coloración natural del cáliz como defectos superficiales. Esto elevó artificialmente el porcentaje de superficie afectada, reclasificando manzanas sanas como Categoría II o Categoría III.
- Categoría II (25%): sólo 1 de 4 manzanas fue clasificada correctamente. Los 3 restantes presentaban defectos cuyo color era suficientemente diferente para ser detectado, pero el porcentaje calculado superaba el umbral de Categoría II, enviándolas a Descarte o Categoría III.
- Categoría III (0%): las manzanas con defectos moderados no fueron detectadas correctamente. Los defectos superficiales de estas manzanas (cicatrices, manchas leves) tenían valores HSV similares al color natural de la epidermis roja, resultandos indetectables con rangos de color.

- Descarte (0%): los defectos graves tampoco fueron detectados correctamente por la misma razón: las cicatrices y heridas presentaban valores H entre 4 y 12, S entre 140 y 220 y V entre 170 y 230, prácticamente idénticos al rojo natural de la manzana sana.

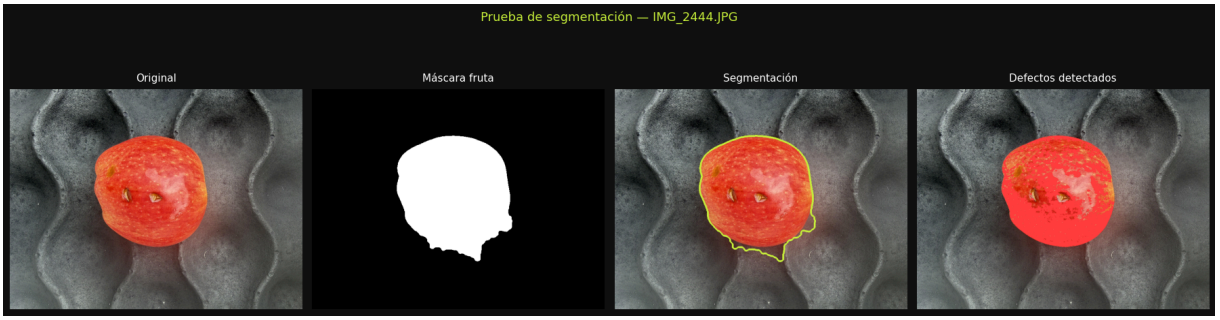


Figura 6.2. Ejemplo de fallo del Método HSV: El sistema sobreestima la superficie afectada al detectar como defecto regiones que no corresponden exclusivamente a lesiones reales, lo que evidencia la presencia de falsos positivos y limita la precisión del método.

Estos resultados confirman la limitación estructural de los métodos basados en reglas de color para este problema específico: cuando los defectos tienen colores similares a la fruta sana, la segmentación HSV no puede distinguirlos independientemente de cuánto se ajusten los rangos de calibración. Esta conclusión motivó directamente la implementación del segundo método basado en Transfer Learning.

6.2 Resultados del Transfer Learning

6.2.1 Proceso de entrenamiento

El modelo MobileNetV2 fue entrenado durante 23 épocas totales (12 en la Fase 1 y 11 en la Fase 2), deteniéndose en ambas fases por Early Stopping al no observarse mejora en val_accuracy durante 8 épocas consecutivas. La siguiente tabla presenta las épocas más relevantes del proceso:

Fase	Época	Train Acc.	Val Acc.	Observación
1	1	30,8%	30,0%	Inicio del entrenamiento
1	4	52,7%	70,0%	Mejor modelo Fase 1
1	12 (final)	67,0%	50,0%	Early stopping activado
2	1	59,3%	70,0%	Inicio fine-tuning

2	3	63,7%	80,0%	Mejor modelo final
2	11 (final)	73,6%	70,0%	Early stopping activado

Tabla 6.2. Resumen del proceso de entrenamiento por épocas relevantes.

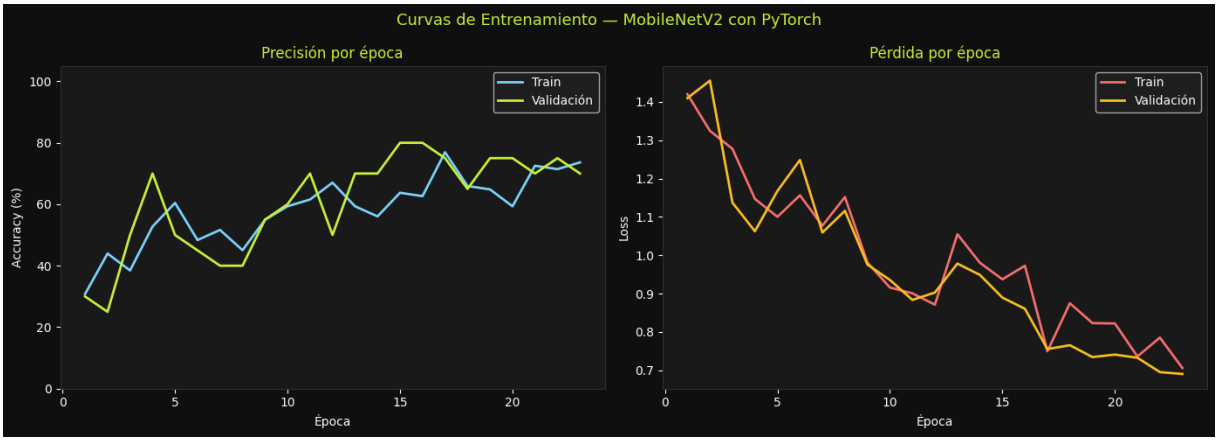


Figura 6.3. Curvas de entrenamiento del modelo MobileNetV2. Izquierda: precisión por época. Derecha: pérdida por época. La pérdida desciende consistentemente de 1,4 a 0,7 a lo largo de las 23 épocas totales.

Las curvas de entrenamiento muestran un comportamiento consistente con el tamaño reducido del dataset. La precisión de validación fluctúa entre 40% y 80% a lo largo del proceso, lo que resulta esperable dado que el conjunto de validación contiene solo 20 imágenes: una predicción incorrecta representa una variación de 5 puntos porcentuales. Aun así, se observa una tendencia general ascendente en la precisión y una disminución sostenida de la pérdida, lo que sugiere que el modelo logró aprender patrones relevantes sin evidenciar un sobreajuste severo.

6.2.2 Métricas de evaluación

El modelo final fue evaluado sobre el conjunto de validación compuesto por 20 imágenes, distribuidas uniformemente entre las cuatro categorías consideradas. Las métricas se calcularon mediante la librería scikit-learn y se presentan en la Tabla 6.3.

Categoría	Precisión	Recall	F1-score	Support	Precisión por cat.
Extra	0,67	0,80	0,73	5	80%
Categoría II	1,00	0,40	0,57	5	40%

Categoría III	1,00	1,00	1,00	5	100%
Descarte	0,71	1,00	0,83	5	100%
ACCURACY GENERAL			0,80	20	80%

Tabla 6.3. Métricas de evaluación del modelo Transfer Learning sobre el conjunto de validación.

Las métricas se interpretan de la siguiente manera:

- Precisión (precisión): de todas las imágenes que el modelo predijo como categoría X, ¿qué porcentaje realmente lo era? Por ejemplo, Categoría II tiene precisión = 1,00: cada vez que el modelo predijo Categoría II, acertó. Sin embargo, no siempre detectó todas las Categoría II (recall bajo).
- Recall (sensibilidad): de todas las imágenes que realmente eran categoría X, ¿qué porcentaje el modelo identificó correctamente? Descarte tiene recall = 1,00: el modelo encontró el 100% de las manzanas de Descarte. Extra tiene recall = 0,80: identificó 4 de 5 manzanas Extra correctamente.
- F1-score: promedio armónico entre precisión y recall. Valor entre 0 y 1, siendo 1 el resultado perfecto. Categoría III alcanza F1 = 1,00 (clasificación perfecta en ambas métricas). Categoría II tiene F1 = 0,57, reflejando la dificultad del modelo para distinguir esta categoría intermedia.
- Support: cantidad de imágenes de esa categoría en el conjunto de validación (5 por categoría en todos los casos).

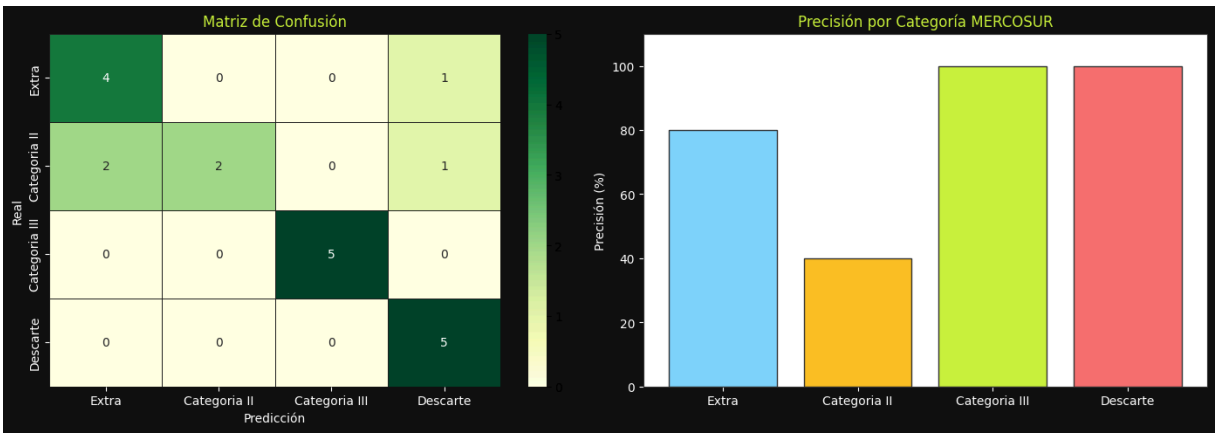


Figura 6.4. Resultados del modelo Transfer Learning: matriz de confusión (izquierda) y precisión por categoría MERCOSUR (derecha).

6.2.3 Análisis de errores

El modelo cometió 4 errores sobre 20 imágenes de validación. El análisis de la matriz de confusión permite identificar el patrón de cada error:

- Extra: Descarte (1 error): una manzana Extra fue clasificada como Descarte. Este error es el más costoso desde el punto de vista comercial, ya que una manzana de alta calidad sería incorrectamente rechazada. El nivel de confianza bajo (alrededor del 50%) en estas predicciones sugiere que el modelo tenía dudas, lo que podría mitigarse con más ejemplos de entrenamiento de la categoría Extra.
- Categoría II: Extra (2 errores): dos manzanas de Categoría II fueron clasificadas como Extra. El modelo subestimó los defectos presentes, lo que sugiere que en esas imágenes los defectos eran leves y visualmente similares a las manzanas Extra del dataset de entrenamiento.
- Categoría II: Descarte (1 error): una manzana de Categoría II fue clasificada como Descarte. El modelo sobreestimó la gravedad de los defectos en este caso.

Los 3 errores en Categoría II (de un total de 4 errores) confirman que esta es la categoría más difícil de clasificar. Su posición intermedia en el espectro de calidad hace que sus características visuales se solapen con las de las categorías adyacentes (Extra y Categoría III), especialmente con un dataset tan reducido (24 imágenes de entrenamiento para esta clase).

6.3 Comparación entre métodos

La siguiente tabla resume la comparación entre los dos métodos implementados. Debe señalarse, sin embargo, que los resultados no se obtuvieron exactamente sobre la misma unidad de análisis: el método HSV fue evaluado a nivel de manzana, mientras que el modelo de Transfer Learning fue evaluado a nivel de imagen sobre el conjunto de validación. Aun con esta diferencia, la comparación permite identificar tendencias claras en términos de desempeño, interpretabilidad y viabilidad práctica.

Aspecto	Método 1: HSV (OpenCV)	Método 2: Transfer Learning (MobileNetV2)
Precisión general	6,2% (1/16 manzanas)	80,0% (16/20 imágenes)
Extra	0%	80%
Categoría II	25%	40%
Categoría III	0%	100%
Descarte	0%	100%
Requiere entrenamiento	No — se basa en reglas manuales	Sí — 23 épocas con GPU

Explicabilidad	Alta — las reglas HSV son visibles e interpretables	Media — red neuronal de caja negra parcial
Tecnología principal	Python, OpenCV, NumPy	Python, PyTorch, MobileNetV2, CUDA
Uso de GPU	No requerida	Sí (NVIDIA RTX 3080, CUDA 12.1)
Tiempo de clasificación	~2–3 segundos por imagen (CPU)	~0,5–1 segundo por imagen (GPU)
Limitación principal	Defectos del mismo color que la fruta sana son indetectables	Dataset pequeño limita la precisión en categorías intermedias
Mejora respecto al anterior	— (método base)	+73,8 puntos porcentuales

Tabla 6.4. Comparación directa entre los métodos HSV y Transfer Learning.

La mejora entre métodos fue de 73,8 puntos porcentuales en precisión general (de 6,2% a 80,0%). Esta diferencia ilustra el salto cualitativo que representa el aprendizaje profundo respecto a los enfoques basados en reglas de color para este tipo de problema, especialmente cuando los defectos son visualmente similares al color natural de la fruta.

El método HSV tiene la ventaja de no requerir entrenamiento ni GPU, y sus reglas son completamente interpretables. Sin embargo, su dependencia de rangos de color lo hace inadecuado cuando los defectos no tienen un color distintivo. El Transfer Learning resuelve esta limitación al aprender características de textura, forma y contexto además del color, pero requiere un proceso de entrenamiento y hardware con GPU para tiempos razonables.

6.4 Validación de la hipótesis

La hipótesis planteada en la Sección 1.4 establece que el sistema permitirá clasificar las manzanas con un nivel de precisión comparable o superior al obtenido mediante la inspección manual, con las siguientes métricas de validación:

Métrica de validación	Meta	Resultado obtenido
Precisión general > 70% sobre conjunto de validación	> 70%	80% (Cumplida)

Clasificación correcta ≥ 90% de los extremos comerciales (Extra y Descarte)	≥ 90%	90% promedio (Extra: 80%, Descarte: 100%) (Cumplida parcialmente)
Costo de implementación de hardware < USD 40	< USD 40	USD 15–40 estimados (Cumplida)

Tabla 6.5. Validación de las métricas de la hipótesis de trabajo.

6.4.1 Análisis de la validación

Los resultados permiten considerar que la hipótesis de trabajo fue validada de manera parcial-favorable:

Métrica 1 - Precisión general:

Se superó el umbral del 70% establecido como meta, alcanzando el 80% de precisión general con el método de Transfer Learning. Esta métrica se cumple satisfactoriamente.

Métrica 2 - Clasificación de extremos comerciales:

Los extremos comerciales del sistema son Extra (la categoría de mayor calidad, sin defectos significativos) y Descarte (la de menor calidad, no apta para comercialización). Descarte alcanzó el 100% de precisión, clasificando correctamente las 5 manzanas del conjunto de validación sin ningún error. Extra alcanzó el 80%: 4 de 5 manzanas fueron clasificadas correctamente, con 1 manzana Extra clasificada incorrectamente como Descarte. El promedio de ambos extremos es del 90%, alcanzando la meta establecida en la hipótesis.

Métrica 3 - Costo de implementación:

El costo estimado del hardware adicional necesario (iluminación LED y bandeja de fondo) se sitúa entre USD 15 y USD 40, cumpliendo el requisito planteado. El software utilizado (Python, PyTorch, OpenCV y todas las librerías relacionadas) es de código abierto y sin costo de licencia.

6.4.2 Limitaciones identificadas

Los resultados obtenidos son consistentes con las limitaciones propias de un prototipo experimental:

- Dataset reducido: con 111 imágenes totales y 20 imágenes de validación (5 por categoría), los resultados presentan variabilidad estadística. Una sola predicción incorrecta equivale a 5 puntos porcentuales en las métricas por categoría. Un dataset más amplio permitiría resultados más estables y confiables.

- Dificultad en Categoría II: la categoría intermedia obtuvo la menor precisión (40%) en ambos métodos. Sus características visuales se solapan con las categorías adyacentes, y el número reducido de ejemplos de entrenamiento no fue suficiente para que el modelo aprendiera a distinguirla con confianza.
- Variedad única: el sistema fue entrenado y evaluado exclusivamente con manzanas de la variedad Red Delicious bajo condiciones controladas. Su aplicación a otras variedades o condiciones de iluminación diferentes requeriría nueva recolección de datos y posiblemente reentrenamiento.
- Adaptación de la normativa: la normativa MERCOSUR define los límites por porcentaje de unidades defectuosas en un lote. La adaptación a clasificación individual por fruto, adoptada en este trabajo, es una simplificación válida para el prototipo que debe tenerse en cuenta al interpretar los resultados en contexto industrial.
- Entorno controlado: la captura de imágenes se realizó bajo condiciones relativamente controladas de fondo, iluminación y posición de la cámara. Por lo tanto, los resultados obtenidos no deben extrapolarse de forma directa a un entorno industrial o a escenarios no controlados sin una validación adicional.

7. Conclusiones

7.1 Generales

El presente trabajo final de carrera demuestra la viabilidad técnica de un sistema de visión artificial para la clasificación automatizada de manzana según las categorías definidas por la normativa MERCOSUR GMC RES. 117/1996, orientado específicamente al segmento de pequeños y medianos productores del Alto Valle del Rio Negro.

Se implementaron y compararon dos enfoques metodológicos de complejidad creciente. El primero, basado en segmentación por colores en el espacio HSV mediante OpenCV, evidencio las limitaciones estructurales de los métodos basados en reglas de color cuando los defectos presentan tonalidades similares a la fruta sana, alcanzando una precisión del 6,2%. El segundo, basado en Transfer Learning con la arquitectura MobileNetV2 implementada con PyTorch, demostró que el aprendizaje profundo supera estas limitaciones al identificar patrones complejos asociados no solo al color sino también a textura, forma y contexto visual, alcanzando el 80% de precisión general sobre el conjunto de validaciones.

La mejora de 73,8 puntos porcentuales entre métodos constituye la evidencia central de este trabajo que es que el tipo de defectos presente en las manzanas evaluadas, los enfoques solo en las reglas de color son insuficientes y el aprendizaje profundo representa una solución técnicamente superior. Esta conclusión es coherente con la literatura científica revisada en el marco teórico y con los resultados reportados por trabajos relacionados tanto en el ámbito iberoamericano como internacional.

Asimismo, el sistema alcanzo un 100% de precisión en las categorías “Categoría III” y “Descarte” y un 80% en la categoría Extra con un costo de hardware estimado entre 150 USD y 40 USD, considerando un esquema de bajo costo con equipamiento convencional. También, estos resultados permiten considerar parcialmente respaldada la hipótesis de trabajo como también se superó el umbral previsto de precisión general y se mantuvo un costo accesible para el contexto productivo analizado. No obstante, estas métricas deben ser interpretados con cautela debido al tamaño limitado del dataset y del conjunto de validación.

7.2 Sobre la investigación y el prototipo desarrollado

El resultado final del prototipo es satisfactorio para el alcance de un trabajo académico experimental. Alcanzar un rendimiento cercano al 80% con un dataset de apenas 111 imágenes y utilizando exclusivamente hardware convencional permite afirmar que el enfoque elegido es técnicamente viable y que la visión artificial puede aportar calor concreto en la clasificación de manzanas por calidad.

Sin embargo, este porcentaje no debe interpretarse como un resultado definitivo ni como una solución lista para ser implementada sin ajustes en un entorno productivo real. El valor de precisión general alcanzado se obtuvo sobre un conjunto de validación de solo 20 imágenes, distribuidas en 5 categorías, por lo que presenta una variabilidad estadística considerable. Bajo estas condiciones, una única predicción incorrecta modifica el resultado global en cinco puntos porcentuales. Por lo tanto, las métricas obtenidas deben leerse como una evidencia preliminar alentadora pero todavía insuficiente para sostener generalizaciones robustas sin nuevas etapas de validación.

La visita técnica realizada a Kleppe S.A. y la observación del sistema Spectrim en funcionamiento permitieron establecer una comparación relevante entre el prototipo desarrollado y los sistemas industriales de referencia. Si bien ambos comparten la misma idea en general, el uso de visión e inteligencia artificiales para la clasificación por imagen, operan en escalas completamente diferentes en términos de infraestructura, velocidad, volumen de datos y grado de automatización. Esta observación refuerza la aceptación del trabajo, ya que muestra que el desafío no radica en la inexistencia de la tecnología, sino en su adaptación a contextos de menor escala y disponibilidad de recursos.

Respecto al futuro del sistema para pequeños productores, se considera que tiene un potencial real como herramienta de apoyo. En su estado actual no reemplaza el criterio del operario experto ni resuelve por si solo el proceso completo de clasificación comercial, pero si pudiera desempeñar un rol útil en tareas de preclasificación, control preliminar de calidad o apoyo a la toma de decisiones. Su principal aporte potencial radica en objetivar, registrar y hacer más consistente una tarea que hoy depende en gran medida de la observación subjetiva.

7.3 Sobre el proceso de aprendizaje

Uno de los aprendizajes más relevantes surgido del desarrollo de este trabajo fue comprender la diferencia entre implementar una técnica de manera aislada y construir un prototipo funcional con coherencia metodológica de punta a punta. El principal desafío no estuvo únicamente en entrenar modelos o ajustar parámetros, sino en articular de forma consistente todas las etapas del proceso, desde la captura de imágenes, la organización del dataset, definición de categorías, implementación de métodos y análisis crítico de los resultados obtenidos.

El desarrollo del método basado en HSV resultó especialmente valioso desde el punto de vista formativo. Su implementación permitió profundizar en conceptos fundamentales del procesamiento digital de imágenes, como el espacio de color HSV, la influencia de la iluminación y del fondo sobre los valores de cada píxel y el papel de las operaciones morfológicas en la limpieza de segmentaciones. Aunque su desempeño final fue bajo, el trabajo con este enfoque no carecio de valor. Permitió comprender de manera concreta los límites que presentan los métodos basados en reglas cuando el problema involucra ambigüedad visual y alta variabilidad superficial.

El paso hacia Transfer Learning representó un cambio metodológico importante. El uso de un modelo preentrenado permitió comprobar en la práctica cómo una red neuronal convolucional puede capturar patrones visuales más complejos que los derivados de reglas simples de color. Diferencias sutiles asociadas a textura, forma y distribución del defecto en la superficie de la fruta pudieron ser abordadas con mayor eficacia mediante este enfoque, lo que explica en buena medida la mejora obtenida respecto del método inicial.

Otro aprendizaje significativo fue advertir que el rendimiento de un sistema de inteligencia artificial depende no solo de la arquitectura empleada, sino también de la calidad del dataset, de la consistencia del etiquetado, de la preparación de los datos y de la interpretación adecuada de las métricas. Este aspecto, que en la teoría puede parecer secundario frente al modelo, resultó central durante el desarrollo del trabajo. La experiencia permitió constatar que un modelo sofisticado no compensa automáticamente un conjunto de datos reducido, poco diverso o insuficientemente estabilizado.

En resumen, el TFC permitió consolidar conocimientos en visión artificial, procesamiento de imágenes y aprendizaje profundo, pero también desarrollar una mirada más crítica sobre las condiciones necesarias para trasladar un experimento académico a una aplicación concreta. Comprender esa distancia entre prueba de concepto y sistema utilizable en campo real constituye, en si mismo, uno de los aportes formativos más valiosos del proceso.

7.4 Limitaciones identificadas

A lo largo del trabajo se identificaron una serie de limitaciones que deben considerarse al interpretar el alcance y la solidez de los resultados obtenidos.

En primer lugar, el tamaño del dataset constituye la limitación principal del estudio. El sistema fue entrenado y evaluado con un total de 111 imágenes. Esta escala experimental resulta suficiente

para una prueba de concepto, pero no para obtener métricas plenamente estabilizadas ni para sostener conclusiones de alta generalización. En tareas de clasificación con Transfer Learning, la información muestra de manera consistente que conjuntos de datos más amplios y diversos tienden a producir resultados más robustos y estables.

En segundo lugar, todas las imágenes utilizadas corresponden a manzanas de una única variedad, Red Delicious, capturadas bajo condiciones controladas. En consecuencia, el sistema no fue validado sobre otras variedades de relevancia comercial, como Golden Delicious, Granny Smith, Fuji o Gale, ni sobre escenarios con variaciones significativas de iluminación, fondo o distancia de captura. Por ello, cualquier deducción a otras condiciones requiere nuevas instancias de recolección, etiquetado y evaluación.

Otra limitación importante está dada por la dependencia de un protocolo de captura controlado. El prototipo fue diseñado para operar con fondo oscuro uniforme, iluminación difusa constante y distancia relativamente fija entre cámara y fruta. Estas condiciones favorecieron la consistencia experimental, pero restringen la aplicabilidad directa del sistema en contextos no controlados. Variaciones en el entorno de captura podrían afectar tanto la segmentación como el desempeño general del clasificador.

También se identificó una dificultad particular en la clasificación de la Categoria II, que obtuvo la menor precisión entre las clases evaluadas. Esto sugiere una superposición visual relevante con las otras categorías similares y pone en evidencia que las clases intermedias resultan más difíciles de distinguir cuando se trabaja con pocos ejemplos de entrenamiento. Además, esta dificultad podría estar asociada no solo al modelo, sino también a la complejidad propia del criterio de clasificación en esa categoría.

Algo que se debe señalar es que la normativa MERCOSUR utilizada como referencia fue originalmente concebida para establecer tolerancias sobre lotes de fruta y no para clasificar unidades individuales. La adaptación realizada en este trabajo a una clasificación por fruto constituye una simplificación metodológica válida para el desarrollo del prototipo, pero limita la equivalencia directa entre los resultados obtenidos y una aplicación estricta de la normativa en contextos comerciales reales.

Por otra parte, el sistema analiza exclusivamente defectos superficiales visibles en las imágenes capturadas. No permite detectar defectos internos ni alteraciones no perceptibles externamente, como daños fisiológicos o defectos asociados a conservación. Esta limitación no responde únicamente al modelo empleado, sino al propio alcance de cualquier sistema de visión superficial convencional.

Finalmente, la implementación actual requiere conocimientos básicos de Python y del entorno Jupyter Notebook, lo que restringe su uso a perfiles con cierta formación técnica. Para que sea mucho más fácil para la gente que no tiene experiencia en la informática es necesario desarrollar una interfaz gráfica simple y adecuada al contexto de uso.

7.5 Líneas futuras de investigación

A partir de los resultados obtenidos y de las limitaciones identificadas, es posibles proponer diversas líneas de continuidad orientadas a fortalecer la solidez metodológica del sistema y mejorar su potencial de aplicación.

La primera y más importante consiste en la ampliación del dataset. Incrementar la cantidad de imágenes por categorías, incorporar varias variedades de manzana y registrar capturas bajo distintas condiciones de iluminación permitiría mejorar la capacidad de generalización del modelo y obtener métricas más estables. Esta línea representa, probablemente, la mejora de mayor impacto potencial sobre el desempeño global del sistema.

Otra mejora consiste en evaluar arquitecturas alternativas a MobileNetV2, como ResNet50 o EfficientNet-B0, que han sido reportadas en la literatura como opciones relevantes para tareas de clasificación visual. No se trata de asumir de antemano que superaran al modelo actual en este caso particular, sino de ampliar la comparación experimental sobre una base de datos más robusta y representativa.

En tercer lugar, resultaría valioso automatizar el proceso de captura mediante un sistema de rotación controlada que permite obtener imágenes desde ángulos fijos y reproducibles. Una plataforma giratoria motorizada de bajo costo podría reducir la variabilidad introducida por la manipulación manual y mejorar la consistencia del dataset.

También se plantea como línea futura el desarrollo de una interfaz gráfica de usuario que permita utilizar el sistema sin necesidad de ejecutar código. Una aplicación sencilla, ya que sea de escritorio o web, facilitaría el uso del prototipo por parte de operarios o técnicos del sector frutícola sin formación específica en programación.

Finalmente, una línea complementaria de desarrollo consistiría en integrar los resultados de clasificación con sistemas de trazabilidad que permitan registrar origen, lote, fecha de captura y destino comercial, fortaleciendo así la capacidad de documentación objetiva del proceso de control de calidad.

En síntesis, el presente trabajo permitió cumplir los objetivos centrales del TFC en términos de diseño, implementación y evaluación preliminar de un sistema automatizado de clasificación de calidad para manzanas a bajo costo. Si bien los resultados obtenidos no habilitan todavía una adopción productiva directa, si constituyen una evidencia favorable de factibilidad técnica y establecen una base metodológica concreta para futuras etapas de validación, ampliación y mejora del sistema.

8. Bibliografía

[1] I. Rojas Santelices, S. Cano, F. Moreira y Á. Peña-Fritz, “Artificial Vision Systems for Fruit Inspection and Classification: Systematic Literature Review,” *Sensors*, vol. 25, no. 5, art. no. 1524, 2025, doi: 10.3390/s25051524. [En línea]. Disponible en: https://doi.org/10.3390/s25051524

[2] D. H. Ballard y C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1982.

[3] R. C. Gonzalez y R. E. Woods, *Digital Image Processing*, 4th ed. New York, NY, USA: Pearson, 2018.

[4] B. V. Granados-Vega, A. de Jesús Calderón, M. Martínez-Zarco, A. A. Vázquez-Arellano, R. J. Cabral-Rosales, J. I. de la Rosa, I. Torres-Pacheco y R. G. Guevara-González, “Development of a Low-Cost Artificial Vision System as an Alternative for the Automatic Classification of Persian Lemon,” *Foods*, vol. 12, no. 20, art. no. 3829, 2023, doi: 10.3390/foods12203829. [En línea]. Disponible en: https://www.mdpi.com/2304-8158/12/20/3829

[5] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [En línea]. Disponible en: https://www.deeplearningbook.org/

[6] Y. LeCun, Y. Bengio y G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539. [En línea]. Disponible en: https://doi.org/10.1038/nature14539

[7] S. J. Pan y Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191. [En línea]. Disponible en: https://doi.org/10.1109/TKDE.2009.191

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474. [En línea]. Disponible en: https://doi.org/10.1109/CVPR.2018.00474

[9] S. Kornblith, J. Shlens y Q. V. Le, “Do Better ImageNet Models Transfer Better?,” en *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 2661–2671, doi: 10.1109/CVPR.2019.00277. [En línea]. Disponible en: https://doi.org/10.1109/CVPR.2019.00277

[10] C. Shorten y T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, art. no. 60, 2019, doi: 10.1186/s40537-019-0197-0. [En línea]. Disponible en: https://doi.org/10.1186/s40537-019-0197-0

[11] E. L. Pencue y J. León-Téllez, “Detección y clasificación de defectos en frutas mediante el procesamiento digital de imágenes,” *Revista Colombiana de Física*, vol. 35, no. 1, pp. 148–151, 2003. [En línea]. Disponible en: https://www.researchgate.net/publication/238095132_Deteccion_y_clasificacion_de_defectos_en_frutas_mediante_el_procesamiento_digital_de_imagenes

[12] J. C. Olguín-Rojas, J. I. Vasquez-Gomez, G. de J. López-Canteñs y J. C. Herrera-Lozada, “Clasificación de manzanas con redes neuronales convolucionales,” *Revista Fitotecnia Mexicana*, vol. 45, no. 3, pp. 369–378, 2022, doi: 10.35196/rfm.2022.3.369. [En línea]. Disponible en: https://www.researchgate.net/publication/363724531_CLASIFICACION_DE_MANZANAS_CON_REDES_NEURONALES_CONVOLUCIONALES

[13] J. V. Aguilar-Alvarado y M. A. Campoverde-Molina, “Clasificación de frutas basadas en redes neuronales convolucionales,” *Polo del Conocimiento*, vol. 5, no. 1, pp. 3–22, 2020, doi: 10.23857/pc.v5i01.1210. [En línea]. Disponible en: https://polodelconocimiento.com/ojs/index.php/es/article/view/1210

[14] S. Fan, J. Li, Y. Zhang, X. Tian, Q. Wang, X. He, C. Zhang y W. Huang, “On line detection of defective apples using computer vision system combined with deep learning methods,” *Journal of Food Engineering*, vol. 286, art. no. 110102, 2020, doi: 10.1016/j.jfoodeng.2020.110102. [En línea]. Disponible en: https://doi.org/10.1016/j.jfoodeng.2020.110102

[15] Y. Li, G. Yao, Z. Cao, C. Li, C. Luo y J. Wang, “Apple quality identification and classification by image processing based on convolutional neural networks,” *Scientific Reports*, vol. 11, art. no. 16618, 2021, doi: 10.1038/s41598-021-96103-2. [En línea]. Disponible en: https://doi.org/10.1038/s41598-021-96103-2

[16] Secretaría de Agricultura, Ganadería y Pesca, *Informe de pera y manzana 2021 con avances 2022*. Buenos Aires, Argentina: Ministerio de Economía, 2022. [En línea]. Disponible en: https://www.argentina.gob.ar/sites/default/files/2021/09/perasymanzanas-sep2022-1.pdf

[17] Servicio Nacional de Sanidad y Calidad Agroalimentaria, *Patagonia Norte – Mensuarios estadísticos*. [En línea]. Disponible en: https://www.argentina.gob.ar/senasa/patagonia-norte-mensuarios-estadisticos

[18] Grupo Mercado Común del MERCOSUR, *Resolución GMC N.º 117/96: Reglamento técnico MERCOSUR de identidad y calidad de manzana*, 1996. [En línea]. Disponible en: https://www.inmetro.gov.br/barreirastecnicas/PDF/GMC_RES_1996-117.pdf

[19] Secretaría de Agricultura, Ganadería, Pesca y Alimentación, *Resolución SAGPyA N.º 600/97*, 1997. [En línea]. Disponible en: https://servicios.infoleg.gob.ar/infolegInternet/anexos/45000-49999/45467/norma.htm

9. Anexo

9.1 Notebooks de desarrollo

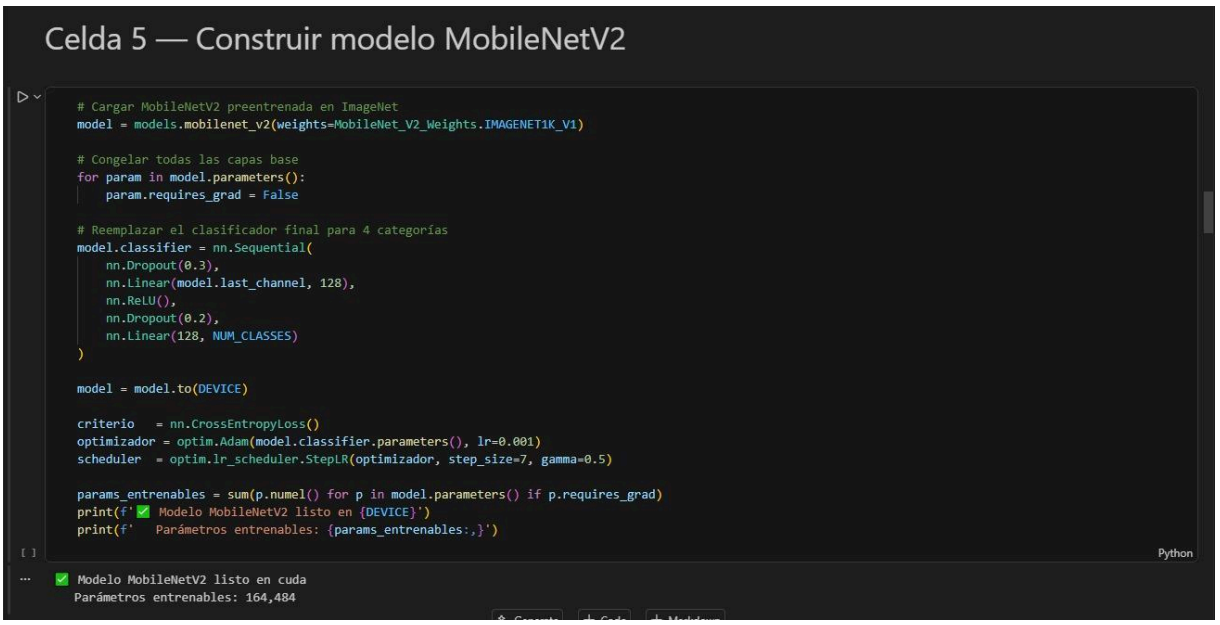


Figura A.1: Construcción del modelo MobileNetV2

La captura muestra el fragmento de código correspondiente a la construcción del modelo de Transfer Learning basado en MobileNetV2. En esta etapa se carga la arquitectura preentrenada y se adapta su clasificador final para resolver el problema específico del trabajo, compuesto por cuatro categorías de calidad: Extra, Categoría II, Categoría III y Descarte.

Celda 4 — Data augmentation y DataLoaders

```python
# Transformaciones con data augmentation para entrenamiento
train_transforms = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.RandomRotation(30),
    transforms.ColorJitter(brightness=0.3, contrast=0.3, saturation=0.2),
    transforms.RandomResizedCrop(IMG_SIZE, scale=(0.8, 1.0)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

# Solo normalización para validación
val_transforms = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

train_dataset = datasets.ImageFolder(DATASET_PATH / 'train', transform=train_transforms)
val_dataset   = datasets.ImageFolder(DATASET_PATH / 'val',   transform=val_transforms)

# WeightedSampler para balancear clases desiguales
class_counts  = np.bincount(train_dataset.targets)
weights       = 1.0 / class_counts[train_dataset.targets]
sampler       = WeightedRandomSampler(weights, len(weights))

train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE, sampler=sampler,  num_workers=0)
val_loader   = DataLoader(val_dataset,   batch_size=BATCH_SIZE, shuffle=False, num_workers=0)

CLASS_NAMES = train_dataset.classes
NUM_CLASSES = len(CLASS_NAMES)

print(f'✅ Dataset cargado')
```

Figura A.2. Implementación de data augmentation y preparación de DataLoaders.

La imagen presenta el bloque de código utilizado para definir las transformaciones aplicadas al conjunto de entrenamiento y para configurar los DataLoaders del modelo. Estas operaciones permitieron ampliar artificialmente la variabilidad del dataset y mejorar la capacidad de generalización del sistema durante el entrenamiento.

Celda 7 — Fine-tuning Fase 2 (descongelar últimas capas)

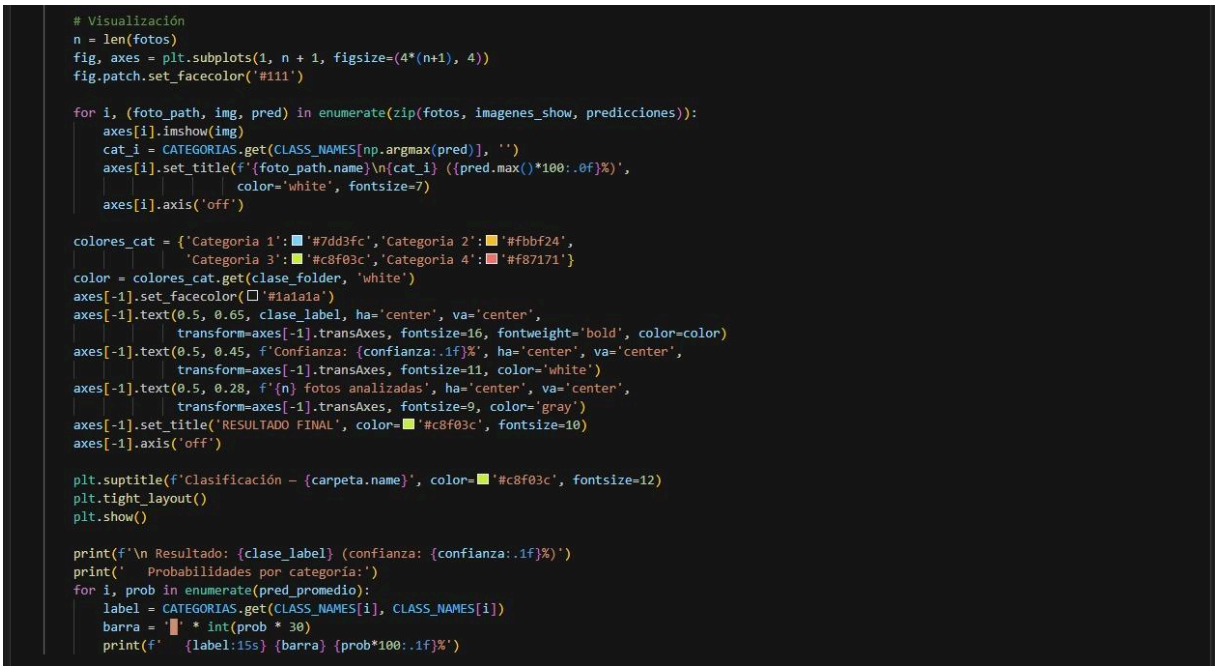
```python
# Cargar el mejor modelo de la Fase 1
model.load_state_dict(torch.load(MODEL_PATH, map_location=DEVICE))

# Descongelar las últimas capas de MobileNetV2
for param in model.features[-3:].parameters():
    param.requires_grad = True

# Learning rate más bajo para fine-tuning
optimizador2 = optim.Adam([
    {'params': model.features[-3:].parameters(), 'lr': 0.00005},
    {'params': model.classifier.parameters(),    'lr': 0.0001}
])
scheduler2 = optim.lr_scheduler.StepLR(optimizador2, step_size=5, gamma=0.5)

paciencia2 = 0
print('🚀 Fase 2 — Fine-tuning últimas capas...\n')

for epoch in range(20):
    t_loss, t_acc = entrenar_epoch(model, train_loader, criterio, optimizador2)
    v_loss, v_acc = evaluar(model, val_loader, criterio)
    scheduler2.step()

    historia['train_loss'].append(t_loss)
    historia['val_loss'].append(v_loss)
    historia['train_acc'].append(t_acc)
    historia['val_acc'].append(v_acc)

    mejoro = '★' if v_acc > mejor_acc else ' '
    print(f'Época {epoch+1:02d}/20 | Train: {t_acc*100:.1f}% | Val: {v_acc*100:.1f}% {mejoro}')
```

Figura A.3. Implementación del proceso de fine-tuning sobre MobileNetV2.

La captura muestra el fragmento de código correspondiente a la segunda fase de entrenamiento del modelo, en la que se cargan los pesos obtenidos en la fase inicial, se descongelan parcialmente las últimas capas de MobileNetV2 y se realiza un ajuste fino con una tasa de aprendizaje reducida. Esta etapa permitió adaptar con mayor precisión las representaciones aprendidas al problema específico de clasificación de manzanas.

Celda 10 — Clasificar una manzana nueva (múltiples fotos → 1 resultado)

```python
def clasificar_manzana_nueva(carpeta_manzana):
    """
    Recibe una carpeta con todas las fotos de UNA manzana.
    Promedia las predicciones y devuelve la categoría final.
    """
    carpeta = Path(carpeta_manzana)
    extensiones = {'.jpg', '.jpeg', '.png', '.JPG', '.JPEG', '.PNG'}
    fotos = sorted([f for f in carpeta.iterdir() if f.suffix in extensiones])

    if not fotos:
        print('No se encontraron fotos')
        return

    model.load_state_dict(torch.load(MODEL_PATH, map_location=DEVICE))
    model.eval()

    predicciones = []
    imagenes_show = []

    with torch.no_grad():
        for foto_path in fotos:
            img = Image.open(foto_path).convert('RGB')
            imagenes_show.append(img)
            tensor = val_transforms(img).unsqueeze(0).to(DEVICE)
            probs  = torch.softmax(model(tensor), dim=1).cpu().numpy()[0]
            predicciones.append(probs)

    pred_promedio = np.mean(predicciones, axis=0)
    clase_idx     = np.argmax(pred_promedio)
    clase_folder  = CLASS_NAMES[clase_idx]
    clase_label   = CATEGORIAS.get(clase_folder, clase_folder)
    confianza     = pred_promedio[clase_idx] * 100
```



```python
# Visualización
n = len(fotos)
fig, axes = plt.subplots(1, n + 1, figsize=(4*(n+1), 4))
fig.patch.set_facecolor('#111')

for i, (foto_path, img, pred) in enumerate(zip(fotos, imagenes_show, predicciones)):
    axes[i].imshow(img)
    cat_i = CATEGORIAS.get(CLASS_NAMES[np.argmax(pred)], '')
    axes[i].set_title(f'{foto_path.name}\n{cat_i} ({pred.max()*100:.0f}%)',
                      color='white', fontsize=7)
    axes[i].axis('off')

colores_cat = {'Categoria 1':'#7dd3fc','Categoria 2':'#fbbf24',
               'Categoria 3':'#c8f03c','Categoria 4':'#f87171'}
color = colores_cat.get(clase_folder, 'white')
axes[-1].set_facecolor('#1a1a1a')
axes[-1].text(0.5, 0.65, clase_label, ha='center', va='center',
              transform=axes[-1].transAxes, fontsize=16, fontweight='bold', color=color)
axes[-1].text(0.5, 0.45, f'Confianza: {confianza:.1f}%', ha='center', va='center',
              transform=axes[-1].transAxes, fontsize=11, color='white')
axes[-1].text(0.5, 0.28, f'{n} fotos analizadas', ha='center', va='center',
              transform=axes[-1].transAxes, fontsize=9, color='gray')
axes[-1].set_title('RESULTADO FINAL', color='#c8f03c', fontsize=10)
axes[-1].axis('off')

plt.suptitle(f'Clasificación — {carpeta.name}', color='#c8f03c', fontsize=12)
plt.tight_layout()
plt.show()

print(f'\n Resultado: {clase_label} (confianza: {confianza:.1f}%)')
print('   Probabilidades por categoría:')
for i, prob in enumerate(pred_promedio):
    label = CATEGORIAS.get(CLASS_NAMES[i], CLASS_NAMES[i])
    barra = '' * int(prob * 30)
    print(f'   {label:15s} {barra} {prob*100:.1f}%')
```

Figura A.4. Implementación de la clasificación final por múltiples vistas de una misma manzana.

Las capturas muestran el procedimiento desarrollado para clasificar una manzana a partir de varias imágenes de entrada. En la primera parte se observa la carga de fotografías, la inferencia individual sobre cada una y el cálculo del promedio de probabilidades. En la segunda parte se presenta la visualización del resultado final, incluyendo la categoría asignada, el nivel de confianza y el detalle de las probabilidades obtenidas.

9.2 Notebook del método HSV

Celda 3 — Configuración: rutas y parámetros

```
DATASET_PATH = Path('C:\\Users\\zackc\\OneDrive\\Desktop\\Manzana')

CATEGORIAS = {
    'Categoria 1': {'label': 'Extra', 'color': '#c8f03c'},
    'Categoria 2': {'label': 'Categoria I',    'color': '#7dd3fc'},
    'Categoria 3': {'label': 'Categoria II',   'color': '#fbbf24'},
    'Categoria 4': {'label': 'Descarte',       'color': '#f87171'},
}

# RANGOS HSV PARA DETECCIÓN DE DEFECTOS

HSV_RANGES = {
    'machucon': {
        'lower': np.array([8,  80, 40]),
        'upper': np.array([22, 220, 130]),
        'limite': {'extra': 2, 'categoria_1': 5, 'categoria_2': 8}
    },
    'quemado_sol': {
        'lower': np.array([0,  0,  15]),
        'upper': np.array([180, 50, 70]),
        'limite': {'extra': 1, 'categoria_1': 3, 'categoria_2': 6}
    },
    'herida': {
        'lower': np.array([35, 40, 15]),
        'upper': np.array([80, 150, 90]),
        'limite': {'extra': 1, 'categoria_1': 3, 'categoria_2': 6}
    },
    'sarna': {
        'lower': np.array([20, 40, 15]),
        'upper': np.array([50, 130, 80]),
        'limite': {'extra': 1, 'categoria_1': 3, 'categoria_2': 5}
    },
```

```
        'mancha_amarilla': {
            'lower': np.array([18, 100, 150]),
            'upper': np.array([35, 200, 220]),
            'limite': {'extra': 2, 'categoria_1': 8, 'categoria_2': 15}
        },
        'cicatriz': {
        'lower': np.array([0,  140, 170]),
        'upper': np.array([12, 220, 240]),
        'limite': {'extra': 1, 'categoria_1': 3, 'categoria_2': 6}
    },
    }

    # UMBRALES DE CLASIFICACIÓN FINAL

    LIMITES_CATEGORIA = {
        "extra":      {"graves": 2,  "total": 5},
        "categoria_1": {"graves": 6,  "total": 12},
        "categoria_2": {"graves": 10, "total": 20},
    }

    print("Configuración cargada con valores reales MERCOSUR GMC N°117/1996")
    print(f"   Dataset: {DATASET_PATH.resolve()}")
```

Figura A.5. Configuración general del método HSV.

Las capturas muestran el bloque de código donde se definen rutas, categorías y parámetros base utilizados en la implementación del método de segmentación por color en el espacio HSV.

Celda 4 — Funciones principales de procesamiento

```
def segmentar_fruta(img_bgr):

    #Segmenta la manzana del fondo oscuro.
    #Devuelve la máscara binaria de la fruta.

    hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)
    rojo1   = cv2.inRange(hsv, np.array([0,   100, 80]),  np.array([10,  255, 255]))
    rojo2   = cv2.inRange(hsv, np.array([165, 100, 80]),  np.array([180, 255, 255]))
    naranja = cv2.inRange(hsv, np.array([10,  100, 80]),  np.array([25,  255, 255]))
    verde   = cv2.inRange(hsv, np.array([30,   60, 60]),  np.array([90,  200, 180]))
    amarillo= cv2.inRange(hsv, np.array([20,   80, 120]), np.array([32,  255, 255]))
    mascara = cv2.bitwise_or(rojo1, rojo2)
    mascara = cv2.bitwise_or(mascara, naranja)
    mascara = cv2.bitwise_or(mascara, verde)
    mascara = cv2.bitwise_or(mascara, amarillo)
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15, 15))
    mascara = cv2.morphologyEx(mascara, cv2.MORPH_CLOSE, kernel)
    mascara = cv2.morphologyEx(mascara, cv2.MORPH_OPEN, cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (12, 12)))
    contornos, _ = cv2.findContours(mascara, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    if contornos:
        mayor = max(contornos, key=cv2.contourArea)
        mascara_final = np.zeros_like(mascara)
        cv2.drawContours(mascara_final, [mayor], -1, 255, -1)
        return mascara_final, mayor
    return mascara, None
```

```python
def detectar_defectos(img_bgr, mascara_fruta):

    #Detecta defectos dentro de la máscara de la fruta.
    #Devuelve porcentaje de superficie afectada por cada defecto.

    hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)
    area_fruta = np.sum(mascara_fruta > 0)
    if area_fruta == 0:
        return {d: {"porcentaje": 0.0, "mascara": np.zeros(img_bgr.shape[:2], dtype=np.uint8)} for d in HSV_RANGES}, np.zeros(img_bgr.shape[:2], dtype=np.uint8)
    resultados = {}
    mascara_defectos_total = np.zeros(img_bgr.shape[:2], dtype=np.uint8)
    for nombre, params in HSV_RANGES.items():
        mascara_color  = cv2.inRange(hsv, params["lower"], params["upper"])
        mascara_defecto = cv2.bitwise_and(mascara_color, mascara_fruta)
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
        mascara_defecto = cv2.morphologyEx(mascara_defecto, cv2.MORPH_OPEN, kernel)
        porcentaje = np.sum(mascara_defecto > 0) / area_fruta * 100
        resultados[nombre] = {"porcentaje": porcentaje, "mascara": mascara_defecto}
        mascara_defectos_total = cv2.bitwise_or(mascara_defectos_total, mascara_defecto)
    return resultados, mascara_defectos_total
```

```python
def clasificar_manzana(resultados_por_foto):

    #Clasifica una manzana individual según MERCOSUR GMC N°117/1996.
    #Toma el peor caso visto en cualquiera de las fotos.

    defectos_max = {d: 0.0 for d in HSV_RANGES}
    for r in resultados_por_foto:
        for d, vals in r["defectos"].items():
            if vals["porcentaje"] > defectos_max[d]:
                defectos_max[d] = vals["porcentaje"]

    # Defectos graves: machucon, quemado_sol, herida, sarna
    defectos_graves = ["machucon", "quemado_sol", "herida", "sarna"]
    pct_graves = sum(defectos_max[d] for d in defectos_graves)
    pct_total  = sum(defectos_max.values())

    # Clasificar según los límites de cada categoría
    if pct_graves <= LIMITES_CATEGORIA['extra']['graves'] and pct_total <= LIMITES_CATEGORIA['extra']['total']:
        categoria = 'Categoria 3'
    elif pct_graves <= LIMITES_CATEGORIA['categoria_1']['graves'] and pct_total <= LIMITES_CATEGORIA['categoria_1']['total']:
        categoria = 'Categoria 1'
    elif pct_graves <= LIMITES_CATEGORIA['categoria_2']['graves'] and pct_total <= LIMITES_CATEGORIA['categoria_2']['total']:
        categoria = 'Categoria 2'
    else:
        categoria = 'Categoria 4'

    return categoria, pct_total, pct_graves, defectos_max
```

```python
def cargar_imagen(path):
    #Carga y redimensiona imagen manteniendo proporción.
    img = cv2.imread(str(path))
    if img is None:
        return None
    h, w = img.shape[:2]
    max_dim = 800
    if max(h, w) > max_dim:
        factor = max_dim / max(h, w)
        img = cv2.resize(img, (int(w*factor), int(h*factor)))
    return img


print("Funciones definidas")
```

Figura A.6. Funciones principales de procesamiento del método HSV.

Las imágenes presentan fragmentos de las funciones utilizadas para segmentar la fruta, detectar regiones asociadas a defectos y procesar la información visual necesaria para la clasificación basada en umbrales cromáticos.

9.3 Visita técnica a Kleppe S.A.

Figura A.7. Vista general de la línea de clasificación y empaque observada durante la visita técnica a Kleppe S.A.

La imagen muestra una vista panorámica de la planta de empaque, donde se observan las cintas transportadoras, sectores de circulación interna y parte de la infraestructura asociada al proceso automatizado de clasificación de fruta. Esta fotografía se incorpora con el objetivo de documentar el contexto industrial relevado durante la visita técnica y evidenciar la escala operativa de los sistemas utilizados por grandes empresas frutícolas del Alto Valle del Río Negro.

Figura A.8. Vista superior de la línea automatizada de clasificación observada durante la visita técnica a Kleppe S.A.

La imagen muestra una perspectiva elevada de uno de los sectores principales de la línea de empaque y clasificación de fruta, donde se distinguen las cintas transportadoras, módulos de procesamiento y áreas de circulación interna. Esta fotografía permite apreciar con mayor detalle la escala y el grado de automatización del entorno industrial relevado, que constituye el punto de referencia tecnológico frente al cual se contextualiza el prototipo desarrollado en este trabajo.

9.4 Fotos del dataset

Figura A.9. Bandeja utilizada como fondo oscuro para la captura de imágenes del dataset.

La imagen muestra el soporte físico utilizado durante el protocolo de captura de fotografías de las manzanas. La bandeja, adaptada con una superficie oscura y uniforme, permitió generar contraste respecto del fruto y reducir interferencias del entorno, favoreciendo tanto la segmentación visual como

la estandarización de las condiciones de adquisición.

Figura A.10. Ejemplo de captura individual de una manzana sobre el fondo utilizado para la construcción del dataset.

La imagen muestra una de las fotografías obtenidas durante el proceso de adquisición del dataset, en la que se observa una manzana posicionada sobre la bandeja de fondo oscuro empleada en el protocolo de captura. Este tipo de imagen ilustra las condiciones de contraste, encuadre y disposición utilizadas para mantener la uniformidad visual del conjunto de datos.

10. Glosario

Accuracy (Precisión general)

Proporción total de predicciones correctas realizadas por un modelo sobre el conjunto evaluado.

Adam

Algoritmo de optimización utilizado para ajustar los parámetros de una red neuronal durante el entrenamiento.

Batch size

Cantidad de imágenes procesadas por el modelo antes de realizar una actualización de los pesos.

CNN (Convolutional Neural Network)

Tipo de red neuronal diseñada para el procesamiento y análisis de imágenes.

CrossEntropyLoss

Función de pérdida utilizada en problemas de clasificación multiclase para medir el error entre las predicciones del modelo y las clases reales.

Data augmentation

Conjunto de transformaciones aplicadas a las imágenes de entrenamiento para aumentar artificialmente la variabilidad del dataset.

DataLoader

Estructura utilizada para cargar y organizar los datos en lotes durante el entrenamiento y la validación del modelo.

Dataset

Conjunto de datos utilizado para entrenar, validar o evaluar un modelo.

Deep Learning (Aprendizaje profundo)

Subárea de la inteligencia artificial basada en redes neuronales de múltiples capas capaces de aprender representaciones complejas de los datos.

Dropout

Técnica de regularización que desactiva aleatoriamente neuronas durante el entrenamiento para reducir el sobreajuste.

Early Stopping

Criterio de detención del entrenamiento que finaliza el proceso cuando el modelo deja de mejorar sobre el conjunto de validación.

Época (Epoch)

Ciclo completo en el que el modelo recorre todas las imágenes del conjunto de entrenamiento una vez.

F1-score

Métrica que combina precisión y recall en un único valor, útil para evaluar el equilibrio entre ambas.

Fine-tuning

Estrategia de Transfer Learning que consiste en descongelar y reajustar parcialmente capas de un modelo preentrenado para adaptarlo a una nueva tarea.

Framework

Conjunto de herramientas y bibliotecas que facilitan el desarrollo e implementación de aplicaciones de software o modelos de aprendizaje automático.

GPU (Graphics Processing Unit)

Unidad de procesamiento especializada en operaciones paralelas, utilizada para acelerar el entrenamiento de modelos de deep learning.

HSV (Hue, Saturation, Value)

Espacio de color que representa una imagen según tono, saturación y valor, útil para tareas de segmentación cromática.

ImageFolder

Estructura de carga de datos de PyTorch que organiza automáticamente imágenes en función de carpetas asociadas a cada clase.

ImageNet

Gran base de datos de imágenes etiquetadas utilizada como referencia para el preentrenamiento de modelos de visión artificial.

Inferencia

Proceso mediante el cual un modelo ya entrenado realiza predicciones sobre datos nuevos.

Learning rate (Tasa de aprendizaje)

Parámetro que define el tamaño de los ajustes realizados por el optimizador sobre los pesos del modelo en cada iteración.

Matriz de confusión

Tabla que resume las predicciones correctas e incorrectas de un clasificador por categoría.

MobileNetV2

Arquitectura de red neuronal convolucional optimizada para dispositivos con recursos computacionales limitados.

Modelo preentrenado

Modelo entrenado previamente sobre un gran volumen de datos y reutilizado como base para una nueva tarea.

Normalización

Proceso de ajuste de los valores de entrada para que se encuentren en una escala adecuada para el entrenamiento del modelo.

OpenCV

Biblioteca de programación utilizada para procesamiento digital de imágenes y visión artificial.

Overfitting (Sobreajuste)

Fenómeno en el que un modelo aprende demasiado bien los datos de entrenamiento y pierde capacidad de generalización sobre datos nuevos.

Pooling

Operación que reduce el tamaño espacial de los mapas de activación en una CNN, conservando la información más relevante.

Precisión (Precision)

Proporción de predicciones positivas correctas sobre el total de predicciones realizadas para una clase.

PyTorch

Framework de aprendizaje profundo utilizado para construir, entrenar y evaluar redes neuronales.

Recall (Sensibilidad)

Proporción de casos reales de una clase que el modelo logra identificar correctamente.

Regularización

Conjunto de técnicas utilizadas para reducir el sobreajuste y mejorar la capacidad de generalización de un modelo.

ReLU (Rectified Linear Unit)

Función de activación ampliamente utilizada en redes neuronales, que reemplaza los valores negativos por cero.

Scheduler

Mecanismo que ajusta automáticamente la tasa de aprendizaje a lo largo del entrenamiento.

Scikit-learn

Biblioteca de Python utilizada para evaluación de modelos y cálculo de métricas de clasificación.

Segmentación

Proceso de separar la región de interés de una imagen respecto del fondo o de otras regiones no relevantes.

Softmax

Función de activación utilizada en la capa de salida de problemas multiclase para convertir valores en probabilidades.

Transfer Learning

Técnica que reutiliza el conocimiento aprendido por un modelo en una tarea previa para resolver una nueva tarea relacionada.

Validación

Etapa de evaluación del modelo sobre datos no utilizados en el entrenamiento, destinada a estimar su capacidad de generalización.

WeightedRandomSampler

Mecanismo de muestreo utilizado para balancear la probabilidad de selección de imágenes de distintas clases durante el entrenamiento.