INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN CÓMPUTO

ÁREA: SISTEMAS INTELIGENTES

Sistema mecatrónico para la clasificación y detección de daños en manzana (Malus domestica)

TESIS

Qua para obtener el grado de;

DOCTOR EN INGENIERÍA DE SISTEMAS ROBÓTICOS Y MECATRÓNICOS

PRESENTA:

M.I. JUAN CARLOS OLGUÍN ROJAS

DIRECTORES:

DR. JUAN IRVING VÁSQUEZ GÓMEZ

DR. JUAN CARLOS HERRERA LOZADA

CIUDAD DE MÉXICO

Abril de 2023

SIP-13
REP 2017

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS
Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a 21 de enero del 2022

El Colegio de Profesores de Posgrado de CIDETEC en su Sesión
(Unidad Académica)

ordinaria No. 1 celebrada el día 18 del mes de enero de 2022, conoció la solicitud presentada por el alumno :

Apellido Paterno:	Olguín	Apellido Materno:	Rojas	Nombre (s):	Juan Carlos

Número de registro: B 2 0 0 9 2 6

del Programa Académico de Posgrado: Doctorado en Ingeniería de Sistemas Robóticos y Mecatrónicos

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

Sistema mecatrónico para la clasificación y detección de daños en manzana (Malus domestica)

Objetivo general del trabajo de tesis:

Desarrollar un sistema mecatrónico para la clasificación visual y detección de daños en manzana (Malus domestica) en el espectro visible.

2.- Se designa como Directores de Tesis a los profesores:

Director: Dr. Juan Irving Vásquez Gómez 2° Director: Dr. Juan Carlos Herrera Lozada

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

CIDETEC

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director de Tesis

Dr. Juan Irving Vásquez Gómez

2° Director de Tesis

Dr. Juan Carlos Herrera Lozada

Aspirante

M. en I. Juan Carlos Olguín Rojas

Presidente del Colegio

Dr. Itzamá López Yáñez

S. E. P.
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN COMPUTO

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 10:00 horas del día 13 del mes de junio del 2023 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: CIDETEC para examinar la tesis titulada:

Sistema mecatrónico para la clasificación y detección de daños en manzana (Malus domestica)

del alumno:

Apellido Paterno:	Olguín	Apellido Materno:	Rojas	Nombre (s):	Juan Carlos

Número de registro: B 2 0 0 9 2 6

Aspirante del Programa Académico de Posgrado: Doctorado en Ingeniería de Sistemas Robóticos y Mecatrónicos

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 14 % de similitud. **Se adjunta reporte de software utilizado.**

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** [ ] **NO** [✓] **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN:
El 14 % de similitud encontrado se debe a coincidencias de formato la mayoría menores al 1%.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente, y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** [✓] **SUSPENDER** [ ] **NO APROBAR** [ ] la tesis por **UNANIMIDAD** [✓] o **MAYORÍA** [ ] en virtud de los motivos siguientes:

El documento satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

COMISIÓN REVISORA DE TESIS

Director de Tesis
Dr. Juan Irving Vásquez Gómez

Director de Tesis
Dr. Juan Carlos Herrera Lozada

Dr. Mario Aldape Pérez

Dr. Oscar Octavio Gutiérrez Frías

Dr. Gilberto de Jesús López Canteñs

Dr. Itzamá López Yáñez
PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN COMPUTO

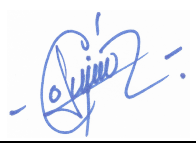
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 14 del mes de junio del año 2023, el (la) que suscribe **Juan Carlos Olguín Rojas** alumno(a) del programa **Doctorado en Ingeniería de Sistemas Robóticos y Mecatrónicos** con número de registro **B200926**, adscrito(a) al **Centro de Innovación y Desarrollo Tecnológico en Cómputo**, manifiesta que es autor(a) intelectual del presente trabajo de tesis bajo la dirección del Dr. Juan Irving Vásquez Gómez y el Dr. Juan Carlos Herrera Lozada y cede los derechos del trabajo intitulado Sistema mecatrónico para la clasificación y detección de daños en manzana (*Malus domestica*), al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o director. Este puede ser obtenido escribiendo a las siguiente(s) dirección(es) de correo jc.olguinr@gmail.com . Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.



Juan Carlos Olguín Rojas

Resumen

En este trabajo de investigación nivel doctorado se realizó el desarrollo de un sistema mecatrónico para la clasificación visual y detección de daños en manzana (Malus domestica) en el espectro visible. Las variedades de manzanas consideradas en la investigación fueron: “Gala”, “Golden”, “Granny Smith” y “Red Delicious”.

Es importante identificar que, dentro del proceso de recolección y la postcosecha de manzanas, es esencial llevar a cabo una clasificación precisa debido a que los frutos son evaluados de acuerdo con su grado de madurez, lo que a se vez determina los precios en el mercado. En la actualidad, tanto en puntos de venta como en empresas agroindustriales de México, la clasificación de las manzanas se realiza manualmente, lo que ocasiona algunas deficiencias respecto a la calidad del producto. No obstante, estos desafíos pueden mitigarse mediante la incorporación de sistemas de visión en sitio, equipados con algoritmos de aprendizaje automático ya que mejoran la clasificación y reducen dichas deficiencias. En esta tesis se presenta el análisis y solución de los problemas de clasificación y detección de manzanas para implementaciones de sistemas de visión en sitio que satisfacen restricciones explicitas en el tiempo de respuesta (tiempo real).

Con el propósito de abordar la tarea de clasificación, se evaluaron diversas estructuras de redes neuronales convolucionales (CNN) y se optó por una arquitectura en particular capaz de distinguir entre manzanas sin daños y manzanas dañadas en la etapa de postcosecha. En la sección “Desarrollo del trabajo de tesis, propuesta de solución para el problema de clasificación”, se presenta una metodología para determinar la mejor arquitectura que resuelve la tarea de clasificación, se utilizó la técnica de búsqueda por rejilla que permitió ajustar gradualmente el espacio de búsqueda a un lote más pequeño de posibles combinaciones. Se llevó a cabo una comparación entre las redes neuronales convolucionales (CNN) LeNet5 y VGG16 en términos de su exactitud. Se realizaron diferentes configuraciones (combinaciones de red e hiperparámetros) para la clasificación del objeto de estudio, evaluando el rendimiento de cada una. Después de las pruebas, se determinó que la mejor configuración fue la utilización de LeNet5 entrenada desde cero con el optimizador RMSProp, logrando una exactitud del 97%.

Para resolver el problema de detección, se desarrolló un diseño de experimento en donde las arquitecturas: YOLOv3-Tiny, YOLOv4-Tiny y YOLOv5-s, se aplicaron como tratamientos. A la arquitectura de mejor desempeño se le realizaron algunas propuestas de modificación con la finalidad de optimizar el desempeño en hardware de bajo costo computacional. Las arquitecturas de detección tienen la desventaja del tamaño de la red, lo que afecta el tiempo de procesamiento necesario para detectar objetos. Esto requiere equipos con alta capacidad de cómputo y la necesidad de unidades GPU para la implementación. Por lo tanto, en este trabajo, el diseño de experimento permitió analizar, comparar y determinar las modificaciones y ajustes necesarios de la arquitectura de mejor desempeño para optimizarla y lograr implementaciones en el sistema mecatrónico para detectar daños en manzana, que cumpla con la restricción del tiempo real. Se encontraron tres arquitecturas optimizadas con valores altos de precisión y sensibilidad, cuyos promedios están ente 94 y 99% respectivamente y que permitieron la implementación en una tarjeta de bajo costo computacional Jeston Xavier NX a una velocidad de 37 fps.

Abstract

In this doctoral level research work, the development of a mechatronic system for visual classification and damage detection in apple (Malus domestica) in the visible spectrum was carried out. The varieties of apples considered in the investigation were: "Gala", "Golden", "Granny Smith" and "Red Delicious".

It is important to identify that, within the harvesting and post-harvest process of apples, it is essential to carry out a precise classification because the fruits are evaluated according to their degree of maturity, which in turn determines the prices in the market. . At present, both in points of sale and in agro-industrial companies in Mexico, the classification of apples is done manually, which causes some deficiencies regarding the quality of the product. However, these challenges can be mitigated by incorporating on-site vision systems, equipped with machine learning algorithms, as they improve classification and reduce such deficiencies. This thesis presents the analysis and solution of block classification and detection problems for on-site vision system implementations that satisfy explicit constraints on response time (real time).

In order to address the classification task, various convolutional neural network (CNN) structures were evaluated and a particular architecture capable of distinguishing between apples without damage and apples damaged in the postharvest stage was chosen. In the section "Development of the thesis work, proposed solution for the classification problem", a methodology is presented to determine the best architecture that solves the classification task, the grid search technique was used, which allowed gradually adjusting the space search to a smaller batch of possible combinations. A comparison was made between the LeNet5 and VGG16 convolutional neural networks (CNNs) in terms of their accuracy. Different configurations (network and hyperparameter combinations) were made for the classification of the object of study, evaluating the performance of each one. After the tests, it was determined that the best configuration was the use of LeNet5 trained from scratch with the RMSProp optimizer, achieving an accuracy of 97%.

To solve the detection problem, an experiment design was developed where the architectures: YOLOv3-Tiny, YOLOv4-Tiny and YOLOv5-s, were applied as treatments. Some modification proposals were made to the architecture with the best performance in order to optimize performance on low-cost hardware. Detection architectures are disadvantaged by the size of the network, which affects the processing time required to detect objects. This requires equipment with high computing power and the need for GPUs for implementation. Therefore, in this work, the design of the experiment allowed to analyze, compare and determine the necessary modifications and adjustments of the architecture with the best performance to optimize it and achieve implementations in the mechatronic system to detect apple damage, which complies with the restriction of the real time. Three optimized architectures with high precision and recall values were found, whose averages are between 94 and 99% respectively and that allowed the implementation in a Jeston Xavier NX low computational cost card at a speed of 37 fps.

Agradecimientos

Agradezco primeramente a Dios, por la oportunidad de vida para realizar este estudio nivel doctorado.

Dedico este trabajo a la memoria de mi madre Nohemi Rojas, quien lamentablemente partió en diciembre de 2020 cuando me encontraba en el primer semestre de este programa y nos tocó enfrentar a la pandemia.

También agradezco a toda mi familia, mi padre Carlos Olguín, a Mónica, Iker, Juan Carlos, Janet, Areli, Abimael, Josué, Federico, Luis, Julieta, Daniel y Samuel, quienes me han apoyado e impulsado a terminar con éxito este programa de estudios.

A los amigos de la UACh, quienes siempre me contagian de ánimo y motivación para seguirme superando, en especial a Canek, Alfonso, Carlos, Ernesto, Swamy y César, por mencionar a algunos.

*Un agradecimiento especial al Centro de Innovación y Desarrollo Tecnológico en Cómputo (**CIDETEC**) del Instituto Politécnico Nacional (**IPN**), a la Universidad Autónoma Chapingo (**UACh**) y al Consejo Nacional de Ciencia y Tecnología (**CONACYT**) por todo el apoyo económico y de infraestructura brindado para la realización de este trabajo doctoral.*

Al apoyo y dirección del Dr. Juan Irving Vásquez Gómez, el Dr. Juan Carlos Herrera Lozada y el Dr. Gilberto de Jesús López Canteñs. Muchas Gracias.

INDICIE GENERAL

INDICIE DE FIGURAS

1 INTRODUCCIÓN

El impacto de esta investigación nivel doctorado es de interés actual por la comunidad científica que desarrolla temas relacionados con la ciencia aplicada y el desarrollo de tecnología en el área agrícola. Actualmente existen problemas en el campo mexicano referidos con elementos de mediciones convencionales basados en técnicas físicoquímicas para determinar el grado de madurez o de inocuidad de un fruto; los métodos convencionales refieren que un tiempo de inspección de un lote de manzanas es de aproximadamente de 8 horas y las técnicas utilizadas son destructivas (Willis *et al*.,1990). La clasificación y detección precisa son elementos esenciales en el proceso de cosecha y postcosecha, ya que diferentes frutas son evaluadas en función de su calidad y los precios de mercado se basan en estas inspecciones. En este trabajo se propuso desarrollar un sistema mecatrónico para la clasificación visual y detección de daños en manzana (*Malus domestica*) en el espectro visible, que permite mejorar la toma de decisiones y el tiempo de inspección, con técnicas no invasivas.

El sistema mecatrónico para la clasificación y detección de daños en manzana, propuesto, mejora la toma de decisiones del experto en las tareas de inspección sanitaría, ya que incrementará la información disponible del producto en cuanto a grado de calidad, y brinda el análisis en volumen por lotes de variedades de manzanas en un menor tiempo, gracias a las herramientas de visión e inteligencia artificial.

La visión computacional se ha desarrollado rápidamente en los últimos años, y el aprendizaje profundo ha demostrado ser una herramienta valiosa en este campo. Los algoritmos de inteligencia artificial pueden analizar imágenes de frutos agrícolas para determinar su tamaño, forma, color, madurez, calidad y defectos. Esto es importante para los agricultores, ya que les permite tomar decisiones informadas sobre cuándo y cómo cosechar, clasificar y distribuir sus productos. Además, la capacidad de procesar grandes cantidades de datos de manera eficiente y precisa es una de las principales razones por las que se utilizan los algoritmos de inteligencia artificial para analizar imágenes RGB de frutos agrícolas.

Sin embargo, una dificultad que presenta esta tecnología es alto costo computacional, pues para aplicaciones de clasificación y detección de daños se requiere contar con computadoras equipadas con unidades de GPU enfocadas a la aceleración de cálculos matemáticos. Específicamente, cuando se hace referencia a las redes neuronales profundas (del inglés “Deep learning”) se comprueba que se utilizan para procesar grandes cantidades de datos y extraer patrones complejos para realizar tareas específicas. Estas redes neuronales tienen muchas capas y miles o millones de neuronas interconectadas, lo que las hace muy poderosas en términos de capacidad de procesamiento y aprendizaje. Pero, para poder entrenar estas redes profundas se requiere de una gran cantidad de datos y un alto poder de cómputo. Por ejemplo, el entrenamiento de redes neuronales profundas implica ajustar los parámetros de la red mediante el uso de técnicas de optimización, como el descenso por gradiente, que requieren múltiples cálculos y actualizaciones de los pesos de las neuronas para minimizar la función de pérdida.

Por esta razón en este trabajo de investigación se pretende dar una alternativa de solución a los problemas de clasificación y detección de manzanas para implementaciones de sistemas de visión en sitio, que puedan satisfacen restricciones explicitas en el tiempo de respuesta (tiempo real).

Este documento de tesis está organizado de la siguiente manera: la sección 1 es la introducción, en la sección 2 se presentan los antecedentes, la sección 3 es el marco teórico, además, en la sección 4 se muestra el desarrollo del trabajo de tesis, y finalmente, la sección 5 son las conclusiones de la tesis.

En la sección de antecedentes se presenta un panorama general del problema de clasificación de manzanas, considerando el grado de calidad del fruto, se formaliza el planteamiento del problema, se presenta la motivación, la justificación y los objetivos del trabajo de tesis. Además, se informa de los trabajos relacionados, consultados y citados en esta investigación.

La sección marco teórico se divide en dos secciones principales, la subsección 3.1 clasificación con redes neuronales convolucionales y la subsección 3.2 detección con redes neuronales convolucionales. En estas subsecciones se explica cómo funcionan las redes neuronales convolucionales para realizar tareas de clasificación y detección de objetos, se analizan los fundamentos matemáticos y las métricas de desempeño tanto para el problema de clasificación como para el problema de detección.

La propuesta de solución se presenta en la sección desarrollo del trabajo de tesis, en donde abordamos los diseños de experimentos tanto para resolver el problema de clasificación como el de detección de manzanas. Se evaluaron distintas arquitecturas de redes neuronales convolucionales (CNN) para la clasificación de manzanas en estado saludable o dañado durante el proceso de postcosecha. De esta manera, se seleccionó una arquitectura que cumplía con este objetivo. Además, se utilizó la técnica de búsqueda por rejilla que permitió ajustar gradualmente el espacio de búsqueda a un lote más pequeño de posibles combinaciones. Se comparó la exactitud de las CNN LeNet5 y VGG16. Finalmente, se informan los resultados experimentales.

En el caso del problema de detección, se desarrolló un diseño de experimento en donde se consideraron como tratamientos a las arquitecturas YOLO en sus versiones: YOLOv3-Tiny, YOLOv4-Tiny y YOLOv5-s. Al tratamiento con mejores resultados, se le realizaron algunas propuestas de modificación con la finalidad de optimizar el desempeño de la arquitectura en hardware de bajo costo computacional. Finalmente, en la sección 5 se presentan las conclusiones del trabajo de tesis.

2 ANTECEDENTES

2.1 Motivación

Los sistemas robóticos y mecatrónicos utilizan diversos métodos para la clasificación de frutos en postcosecha. Estos métodos permiten la detección y clasificación automatizada de los frutos según criterios predefinidos de calidad y madurez. Una herramienta no invasiva para el análisis de frutos es la visión por computadora que incluye el análisis de imágenes y la detección de características visuales, así como la medición de propiedades físicas y químicas de los frutos, como el tamaño, la forma, el color, la textura y la densidad.

Una mala clasificación de la manzana evita cumplir con los estándares de calidad descritos en la norma oficial mexicana (más información de la norma en el anexo “A”) y esto ocasiona que México presente en la actualidad un déficit comercial en la exportación de manzanas, pues en 2018 se importaron 282,756 toneladas con un valor de 264 millones de dólares y se exportaron sólo 766 toneladas con un valor de 1.1 millones de dólares, además, la variación en porcentaje de las exportaciones del fruto en 2018 redujo 17.7% con respecto a las exportaciones de 2017 (SIAP, 2020).

El servicio de información agroalimentaria y pesquera (SIAP, 2020) informa que hay un reducido sector de productores nacionales de aguacate y jitomate con una participación en el mercado internacional de 2,625 toneladas y 2,080 toneladas respectivamente (exportaciones de 2018), que están comenzando a utilizar sistemas de clasificación por visión por computadora para fines de cumplimiento de normas de exportación, lo que les permite realizar inspecciones en masa.

2.2 Antecedentes de los sistemas robóticos en la agricultura

En los últimos años, los avances en robótica y mecatrónica han proporcionado soluciones innovadoras y prometedoras para abordar los desafíos asociados con la clasificación eficiente de frutos en la etapa de postcosecha.

Estos sistemas constan de diferentes componentes y equipos. Suelen incluir brazos robóticos o manipuladores, sistemas de transporte y posicionamiento, sensores y cámaras de visión, sistemas de iluminación y sistemas de control. La integración de estos elementos permite la automatización de las tareas de clasificación, mejorando la precisión y la eficiencia del proceso.

Los algoritmos desempeñan un papel crucial en los sistemas robóticos y mecatrónicos para la clasificación de frutos. Estos algoritmos se encargan de procesar los datos recopilados por los sensores y las cámaras, identificar y extraer características relevantes de los frutos, y tomar decisiones de clasificación basadas en criterios predefinidos. Actualmente, las redes neuronales profundas se utilizan ampliamente para mejorar la precisión y la capacidad de adaptación de los sistemas de clasificación.

Dentro de los sistemas robóticos y mecatrónicos para la clasificación de frutos, se han desarrollado equipos y tecnologías específicas para abordar los desafíos particulares de la postcosecha. Por ejemplo, se han diseñado sistemas de transporte y manipulación especializados para garantizar la integridad de los frutos durante el proceso de clasificación. Asimismo, se han utilizado tecnologías de iluminación adaptativa y filtrado de imágenes para mejorar la detección de características visuales en diferentes condiciones ambientales. Estas tecnologías han permitido una mayor precisión, eficiencia y velocidad en el proceso de clasificación, optimizando la calidad y el valor comercial de los frutos. Además, la automatización de estas tareas ha reducido los costos y la dependencia de la mano de obra humana, al tiempo que proporciona resultados confiables.

A medida que la industria agrícola avanza hacia una mayor producción sostenible y demandas más rigurosas en términos de calidad y seguridad alimentaria, los sistemas robóticos y mecatrónicos tienen un papel más destacado en la clasificación de frutos en postcosecha. El panorama a corto y mediano plazo prevé que estas tecnologías seguirán evolucionando, integrando nuevos avances en inteligencia artificial, visión por computadora y automatización, contribuyendo de esta manera con el desarrollo del sector agrícola.

En este trabajo de tesis nivel doctorado se estudia la clasificación y detección de daños en manzana, y en ese sentido, la planeación agrícola nacional 2017-2030 (SIAP, 2020) indica que en 2019 el consumo de manzanas en México fue de aproximadamente 927,000 toneladas, de las cuales la producción nacional abasteció 716,930 toneladas y se importaron 212,680 toneladas. Es decir, el consumo anual per cápita es de aproximadamente 8 kg.

Datos de 2018 indican que México es el productor de manzanas número 20 a nivel internacional con 659,451 toneladas, lo cual generó ingresos anuales para este sector de 7,782 millones de pesos, sin embargo, el flujo comercial internacional no es favorable para México pues mientras importó en ese año 282,756 toneladas, exportó 766 toneladas. En 2019 el estado de Chihuahua produjo 82% de la producción nacional con aproximadamente 624,696 toneladas, el volumen de producción de las principales entidades federativas productoras de manzanas se muestra en el cuadro1.

El precio medio rural de la manzana en 2019 para el estado de Chihuahua fue de 11,777 pesos por tonelada y debido a que su producción alcanzó aproximadamente 624,696 toneladas, en ese año obtuvieron ingresos de 7 mil 357 millones de pesos.

A nivel mundial en 2019, México fue catalogado como el productor número 23 con una producción de 761,483 toneladas. Esta cifra está muy por debajo de los productores chinos que generaron 86.1 millones de toneladas y los estadounidenses con 5.3 millones de toneladas.

Cuadro 1. Volumen en producción de manzana entidades federativas en México.

Clasificación	Entidad federativa	Región	Volumen (Toneladas)	Variación (%) 2018-2019
	Total Nacional		761,483	15.5
1	Chihuahua	Noreste	624,696	9.7
2	Coahuila	Noreste	47,769	369.9
3	Puebla	Centro	34,933	-2.2
4	Durango	Noreste	21,540	93.3
5	Veracruz	Sursureste	9,248	0.1
6	Zacatecas	Noreste	3,938	-11.3
7	Chiapas	Sursureste	3,195	-5.2
8	Hidalgo	Centro	3,138	-6.4
9	Nuevo León	Noreste	3,124	13.2
10	Oaxaca	Sursureste	2,378	4.3
	Resto		7,525	1.6

El flujo comercial internacional comienza a tener un pequeño incremento, pues en 2019 México registró un incremento del 25% en sus ventas por exportaciones de manzana en comparación con lo registrado en 2018. Para observar este incremento se cita la Figura 2.1.

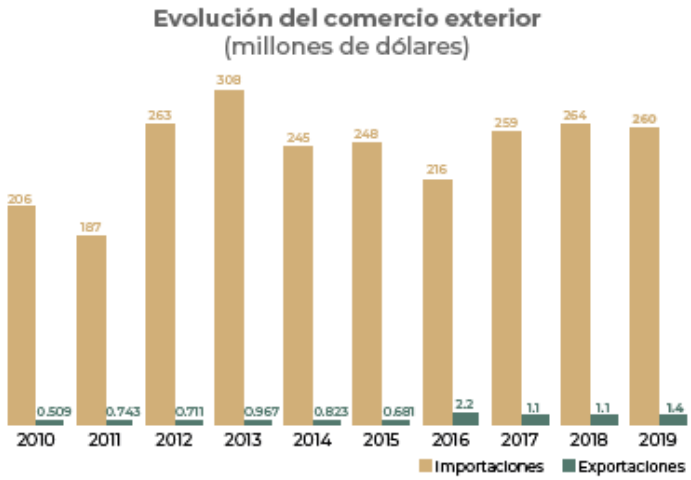


Figura 2.1. Muestra el flujo comercial internacional de la manzana por año para México.

Son varias las razones de las dificultades de exportar la manzana producida en México, pero una se debe a que no se logra cubrir la demanda interna del producto, se requiere mayores superficies cosechadas con mejores rendimientos. Sin embargo, para lograr esto es necesario que los productores cuenten con tecnología de mejoramiento de cosecha y postcosecha, que permitan dar cumplimiento a las normas de calidad de exportación.

2.3 Planteamiento del problema

Actualmente, existen problemas en el campo mexicano referidos con elementos de mediciones convencionales basados en técnicas fisicoquímicas para determinar el grado de calidad de un fruto, principalmente porque estas técnicas requieren instrumental de laboratorio, en la Figura 2.2, se observan los equipos utilizados para esta tarea.

Figura 2.2. Muestra instrumentos de medición utilizados para medir firmeza, el contenido de solidos solubles, la prueba de almidón y la acidez valorable de frutos.

Esto ocasiona que en muchos casos los productores nacionales implementen líneas de inspección visual en donde personas son entrenadas para identificar: daños, grado de madurez, texturas y tamaños de frutos principalmente, sin embargo, estos métodos son subjetivos y dificultan la inspección de grandes lotes del fruto o análisis en masa. Además, los procesos de inspección actuales presentan ciertas limitaciones, ya que requieren de una estructura dedicada exclusivamente a este fin, construida con materiales sólidos y completamente cerrada. Esta planta procesadora debe estar separada de las viviendas familiares y su piso debe ser de concreto para facilitar la limpieza posterior a la selección y el empaque. Además, se requiere un área de al menos 550 metros cuadrados para llevar a cabo estas actividades. Asimismo, es necesario contar con los equipos y dispositivos adecuados para la clasificación y manejo de las frutas, tales como instalaciones eléctricas, tanques de lavado y saneamiento, así como mesas de trabajo y bandas transportadoras (ver Figura 2.3).

Figura 2.3. Muestra los métodos de clasificación de frutos basados en la experiencia y observación.

Los sistemas de visión computacional y que utilizan algoritmos de inteligencia artificial implementados en bandas transportadoras (ver figura 2.4a y 2.4b), son una alternativa eficiente para enfrentar los problemas y dificultades para determinar el grado de calidad en los frutos. Sin embargo, estos sistemas existentes requieren de un edificio (planta procesadora).

Figura 2.4a. Sistemas de clasificación de frutos por visión computacional

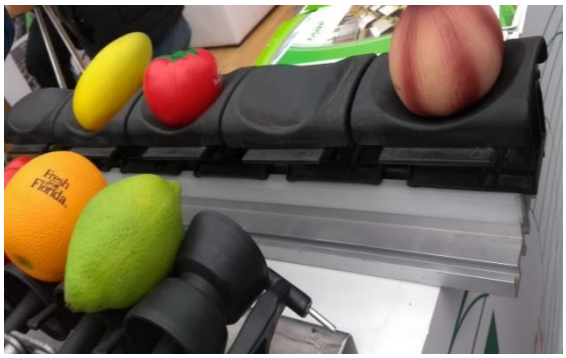


Figura 2.4b. Bandas transportadoras de frutos

Figura 2.4. Sistemas de visión computacional para clasificación de frutos.

A pesar de los avances recientes en visión por computadora para inspección y selección de calidad de manzanas, las redes neuronales convolucionales requieren una gran capacidad de cómputo para realizar la inferencia; por lo tanto, en este trabajo se buscó dar respuesta a ¿Cuál es la arquitectura de red neuronal adecuada para la clasificación de manzanas para un sistema de visión en sitio que cuente con baja capacidad computacional? ¿Cuál es la arquitectura de red adecuada para la detección de manzanas en aplicaciones de tiempo real?

2.4 Justificación

El impacto de esta investigación es de interés actual por la comunidad científica que desarrolla temas relacionados con la ciencia aplicada y el desarrollo de tecnología robótica y mecatrónica para el área agrícola. Los métodos convencionales basados en técnicas fisicoquímicas para determinar la calidad del fruto refieren que un tiempo de inspección de un lote de manzanas es de aproximadamente de 8 horas y las técnicas utilizadas son destructivas (Willis *et al*.,1990), por esta razón se considera importante desarrollar un sistema mecatrónico para la clasificación y detección de daños en manzana que permita mejorar la toma de decisiones, el monitoreo y el tiempo de inspección, con técnicas no invasivas.

Este sistema mecatrónico mejorará la toma de decisiones del experto en las tareas de inspección sanitaría, incrementará la información disponible del producto en cuanto a sus características de calidad, brindará el análisis en volumen por lotes de variedades de manzanas en menor tiempo; por lo que, se propone que el sistema cuente con una serie de sensores y de sistemas embebidos y algoritmos inteligentes para la clasificación de estos lotes. Las arquitecturas de redes neuronales convolucionales de aprendizaje profundo son las que se utilizarán.

2.5 Hipótesis

Se formula que un sistema mecatrónico de visión en sitio desarrollado, acelerará el diagnóstico, clasificación y detección de daños en manzanas, los métodos de visión por computadora implementados en éste trabajo complementarán a los algoritmos de aprendizaje automático para encontrar relaciones y patrones que permitan la correcta identificación y clasificación de características de calidad del fruto, como consecuencia, las arquitecturas óptimas convolucionales desarrolladas en este trabajo podrán ser implementadas en dispositivos embebidos gracias a su bajo costo computacional, y se obtendrá además en el análisis no invasivo en postcosecha, información útil para determinar recomendaciones para el manejo de plagas en el proceso de cultivo.

2.6 Objetivos

2.6.1 Objetivo general

Desarrollar un sistema mecatrónico para la clasificación visual y detección de daños en manzana (Malus domestica).

2.6.2 Objetivos específicos

1) Instrumentar y caracterizar un sistema mecatrónico para la adquisición de información de manzanas en sitio.
2) Determinar la eficiencia de cuatro redes neuronales convolucionales en la clasificación de manzanas.
3) Desarrollar una red neuronal convolucional para detectar en la manzana el daño.
4) Adecuar la arquitectura de una red neuronal para su uso en sistemas de baja capacidad de cómputo.

2.7 Aportes científico y tecnológico

1.1 Aportes científicos

- Un estudio de la eficacia y eficiencia de las redes neuronales convolucionales para clasificación y detección de daños en manzana en sistemas de bajo rendimiento computacional.
- Una arquitectura profunda para detección de daños en manzana, con propuestas de modificación y adecuación a la capacidad de rendimiento del sistema embebido.

1.2 Aportes tecnológicos

- Diseño e instrumentación del sistema mecatrónico para su implementación en sitio.
- Automatización de tareas de supervisión en cadenas de embalaje y selección de manzanas con métodos no invasivos.

2.8 Trabajo relacionado

En la literatura especializada se presentan los trabajos científicos de medición de índices de calidad de frutos, los cuales en su gran mayoría usan técnicas destructivas para realizar una inspección sanitaria (Willis *et al*.,1990). En el trabajo de Crisosto (Crisosto *et al*., 1999) se considera el color de fondo, la firmeza, el contenido en sólidos solubles, la prueba de almidón y la acidez valorable, y se informa que para medir la firmeza de la manzana se utiliza un penetrómetro con un pistón de 11mm. Además, para realizar las medidas físicoquímicas de acidez, la manzana se corta en gajos y también se muele para obtener un zumo, ya que en este proceso es necesario vertir este contenido en vasos de precipitados y pipetas, el procedimiento consiste en añadir 10ml del zumo con 10ml de agua destilada dentro de un vaso de precipitados de 100ml, se introduce la sonda de un PHímetro y se van añadiendo mililitros de Hidróxido de Sodio (NaOH) hasta alcanzar un valor de PH de 8.1, se cuantifican los mililitros de NaOH gastados para realizar el cálculo.

En contraste, la visión artificial ha mostrado eficacia en la clasificación de daños de diversas frutas como se muestra en los siguientes trabajos. En Zhang *et al*., (2017) desarrollaron una cosechadora de manzanas autopropulsada que cuenta con una máquina clasificadora en campo y a través de visión computacional tomaron decisiones de acuerdo con el color, tamaño, forma y defectos. Su sistema mostró bajo costo y rápida velocidad de inspección. Además, en Lu *et al*., (2017) se mejora la detección de los defectos de la manzana al incorporar la superficie del fruto, la presencia de tallos y cálices.

Los métodos que generalmente son utilizados para la clasificación de variedades y detección de defectos superficiales de la manzana “se basan en técnicas de procesamiento de imágenes digitales que consideran la eliminación del fondo, segmentación de defectos y la identificación de las zonas del tallo y del cáliz” (Wang *et al*., 2020).

Sun *et al*., (2017) consideran que un sistema de clasificación automática de manzanas debe involucrar los aspectos de: color, peso, dimensión y defectos. Los autores destacan que el brillo superficial de una manzana refleja la frescura y sus defectos, considerando a esta característica sensorial entre las más importantes. También se reporta el trabajo de Sofu *et al*., (2016) donde encontraron que la característica que más afecta la calidad de la clasificación de la manzana es la mancha y la descomposición.

En el estudio realizado por Moallem y colaboradores (Moallem *et al.*, 2017), se aplicaron técnicas de extracción de características estadísticas, texturales y geométricas a imágenes de manzanas de la variedad Golden Delicious. Utilizando clasificadores como KNN (k-vecinos más cercanos), MLP (perceptrón multicapa) y SVM (máquinas de vectores de soporte), se logró clasificar las manzanas en dos categorías: saludables y dañadas, obteniendo una precisión del 92.5%. Para ilustrar un ejemplo del resultado de las imágenes segmentadas, donde se diferencian los defectos en las manzanas, se cita la Figura 2.5.

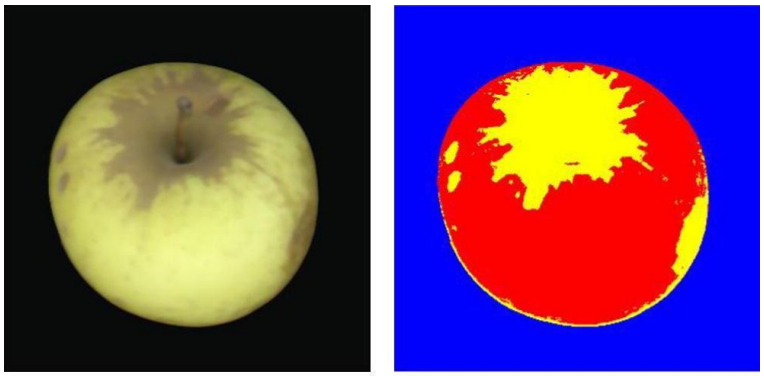


Figura 2.5. Ejemplo de imagen después de eliminar el fondo (a) y el resultado de la segmentación de defectos, de las áreas sanas y del fondo (b) *(Moallem et al., 2017)*.

Además, para asegurar una correcta detección de daños en los sistemas de clasificación, se debe considerar llevar a cabo la singularización, rotación y transporte de las frutas de manera que las manzanas provenientes del transportador principal se alineen con una distancia adecuada entre ellas, y giren continuamente (alrededor del eje del tallo o de forma aleatoria) para permitir la captura de imágenes de toda la superficie de la fruta por parte del sistema de visión por computadora, mientras avanzan simultáneamente (Throop *et al.,* 2001).

En el trabajo reportado por Cubero *et al*. (2011) desarrollaron un sistema de ventosa robótica para sostener y girar frutas individuales debajo de la cámara, permitiendo así la captura de imágenes de la mayor parte de la superficie de la fruta. Sin embargo, se reporta que este diseño tiene un rendimiento bajo y no resulta práctico para su implementación. Posteriormente, Lu *et al*. (2018) desarrollaron un sistema de transporte multifuncional basado en un transportador helicoidal, que logra la separación, rotación y transporte de las frutas de manera conjunta (Figura 2.6).

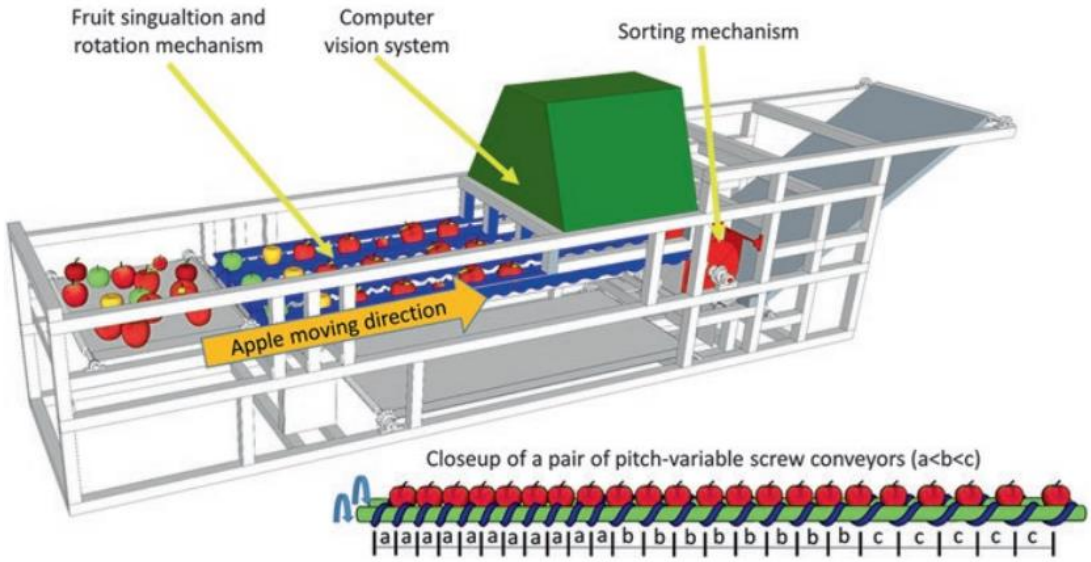


Figura 2.6. Sistema de transporte multifuncional basado en un transportador helicoidal *(Lu et al., 2018)*.

Dentro de la IA están las técnicas de aprendizaje profundo, particularmente las redes neuronales convolucionales (CNN, del inglés *convolutional neural networks*), que aplican múltiples capas de filtros convolucionales de una o más dimensiones a una entrada para la extracción de características y conectadas a una red perceptrón multicapa pueden realizar tareas de clasificación; las entradas generalmente son imágenes. Las redes neuronales convolucionales (CNN) tienen diversas aplicaciones en diferentes campos, incluyendo la agricultura (Kamilaris *et al*., 2018). Se reporta que han sido exitosamente utilizadas para detectar defectos en pepinos (Liu *et al*., 2018) así como en melocotones (Sun *et al.*, 2019). También se ha informado su eficacia en la medición de residuos de plaguicidas en manzanas después de la cosecha (Jiang *et al*., 2019), así como en el porcentaje de floración del algodón (Xu *et al*., 2018) y en la detección de enfermedades en plantas (Barbedo *et al*., 2019). En un estudio realizado por Fan y colaboradores (Fan *et al*., 2020), se entrenó una CNN para clasificar manzanas Gala como sanas o dañadas. El algoritmo fue implementado en una banda transportadora, logrando una exactitud del 92%.

Al respecto Baneh *et al*., (2018) desarrollaron un sistema electrónico que puede clasificar manzanas sobre una banda transportadora e informan de un clasificador neuronal para golden y red delicious que distingue cuatro categorías de acuerdo con la norma europea, obteniendo una exactitud de 89% para golden y 92% para red delicious respectivamente. Wu *et al*., (2020) usaron un modelo de CNN AlexNet modificado con una estructura de 11 capas, con el objetivo de identificar y detectar defectos en las manzanas, utilizaron imágenes de retrodispersión inducidas por láser y alcanzaron una exactitud de 92.5%. Los instrumentos de clasificación mejoran los tiempos de inspección sanitaria, Bhargava *et al*., (2021) afirman que la clasificación simultánea de frutas por tamaño y color ahorraría el tiempo de inspección sanitaria, reduciendo significativamente el manejo de la fruta.

En el trabajo de Duran Melo (Durán *et al*., 2012) reportan acerca de la clasificación de manzanas tomando en cuenta características de su superficie, basándose en un preprocesamiento y segmentación

de la imagen para extraer las características de interés para la clasificación. Por ejemplo, extrajeron 14 características de imágenes de manzanas Gala, tales como: niveles de rojo y verde, contornos internos en la manzana, perímetro del contorno principal, y posteriormente entrenaron una red Perceptrón Multicapa (MLP) para clasificar las manzanas en tres categorías con una exactitud del 88.33%.

Fan *et al*., (2020) usaron una arquitectura de aprendizaje profundo basada en redes neuronales convolucionales (CNN) y un módulo de visión por computadora para detectar manzanas defectuosas en una máquina clasificadora de cuatro líneas a una velocidad de 5 frutas/s. Además, compararon la precisión de la CNN con un método de procesamiento de imágenes basado en el recuento de regiones defectuosas y con un clasificador de máquina de soporte vectorial. La CNN implementada en la máquina clasificadora obtuvo los mejores resultados.

El aprendizaje profundo es uno de los métodos basados en máquina de aprendizaje (ML) más utilizados. Una característica importante del aprendizaje profundo es que tiene altos niveles de abstracción y la capacidad de aprender automáticamente los patrones presentes en las imágenes. En particular, la red neuronal convolucional (CNN) (LeCun *et al*., 1998) es la principal arquitectura de aprendizaje profundo utilizada para el procesamiento de imágenes. (Guo *et al*., 2016). Las CNN son un tipo de redes neuronales artificiales que utilizan operaciones de convolución en al menos una de sus capas. Desde 2012, cuando Krizhevsky *et al.*, (2012) ganó la competencia de clasificación para ImageNet (ILSVRC) (Russakovsky *et al*., 2015), las CNN han ganado gran popularidad como un método eficiente para la clasificación de imágenes en muchos campos. Por ejemplo, en la agricultura, se han utilizado enfoques basados en CNN para la clasificación de frutas (Lu *et al*., 2018) y la detección de frutas (Bargoti *et al*., 2017).

La revisión realizada por Bhargava y Bansal (Bhargava *et al*., 2018) analizan el uso de técnicas de procesamiento de imágenes y visión artificial en la industria agroalimentaria. Definen que las propiedades de calidad más relevantes de los productos agrícolas son el color, el tamaño, la textura, la forma y los defectos. Por lo tanto, los autores presentan una descripción general de los diferentes métodos de preprocesamiento, segmentación, extracción de características y clasificación utilizando estas características. Estudian varios enfoques para la clasificación en la evaluación de la calidad de los alimentos, incluidos KNN, SVM y CNN. Según ellos, los enfoques basados en aprendizaje profundo, como las redes neuronales convolucionales, son muy eficientes para la clasificación y el reconocimiento de frutas, lo que reduce notablemente el error en la clasificación.

Hameed *et al.,* (2018), compararon diferentes métodos de visión por computadora para clasificar frutas y verduras, con SVM, KNN, árboles de decisión, CNN y otros métodos de extracción de características. Además, destacan el hecho de que se han propuesto varios enfoques de clasificación para la evaluación de la calidad y la recolección automática, pero estas técnicas se limitan a unas pocas clases y pequeños conjuntos de datos. Además, su artículo identifica tres grupos principales de aplicaciones de clasificación de frutas y hortalizas: evaluación de calidad, recolección automática e inventario de supermercado.

Los enfoques basados en CNN para tareas de clasificación de frutas se presentan a manera de resumen en el Cuadro 3, en donde se muestran diversos artículos en los que se aplican las CNN para la clasificación de frutas según su tipo. En este caso, se entiende por clasificación el hecho de identificar el tipo de fruta observado en una imagen que contiene un solo tipo o varios tipos de frutas.

Cuadro3. Resumen del estado del arte en tareas de clasificación de frutas con CNN

Autor	Base de imágenes	Datos de entrada	Modelo CNN	Resultados alcanzados
Lu, Y. *et al.*,(2017)	ImageNet	Imagen RGB (128,128,3)	Modelo CNN con 5 capas	74% de exactitud sin aumentación de datos y 90% con aumentación de datos
Zhang, Y. *et al.*,(2019)	VegFru	Imagen RGB (256,256,3)	Modelo CNN con 13 capas	Exactitud de 94.94%
Steinbrener, J. *et al.*,(2019)	Own	Imagen Hiperespectral (256,256,3)	GoogLeNet modificada	88.15% con imágenes pseudo-RGB, 85.93% con combinaciones lineales, 92.23% con núcleo convolucional
Katarzyna, R. *et al.*,(2019)	Own	Imagen RGB (150,150,3)	Modelo CNN con 9 capas	Exactitud del 99.78%
Sakib, S. *et al.*,(2019)	Fruit-360	Imagen RGB (100,100,3)	Modelo CNN propuesto	Exactitud del 100%
Muresan, H. *et al.*,(2017)	Fruit-360	Imagen RGB (100,100,3)	AlexNet, GoogLeNet, y Modelo CNN propuesto	Exactitud de aproximadamente 99% en todos los modelos
Zhu, L. *et al.*, (2018)	ImageNet	Imagen RGB (224,224,3)	AlexNet	Exactitud del 92.1%
Hussain *et al.*,(2018)	Own	Imagen RGB (150,150,3)	Modelo CNN propuesto	Exactitud del 99%
Lu, S. *et al.*,(2018)	Own	Imagen RGB (256,256,3)	Modelo CNN con 6 capas	Exactitud del 91.44%
Patino-Saucedo, A. *et al.*, (2018)	Supesmarket Data	Imagen RGB (48,64,3)	Fruit-AlexNet	Exactitud del 99.54%
Wang, S.H. *et al.*, (2020)	VegFru	Imagen RGB (150,150,3)	Modelo CNN con 8 capas	Exactitud del 95.67%
Zeng, G. *et al.*, (2017)	Own	Imagen RGB (224,224,3)	VGG modificada	Exactitud del 95.6%
Hou, S. *et al.*, (2017)	VegFru	Imagen RGB, tamaño no reportado	CBP-CNN, VGG y Red hibrida.	Exactitud del 87.4%, 84.4% y 88.8% respectivamente
Zhang, W. *et al.*, (2015)	UEC-FOOD100	Imagen RGB (128,128,3)	Modelo CNN con 5 capas	Exactitud del 80.8% para una fruta, y 60.9% en múltiples frutas

Wu *et al*., (2020) tomaron un modelo AlexNet modificado con una estructura de 11 capas, con el objetivo de identificar y detectar defectos en las manzanas. Además, como comparación de la clasificación, utilizan tres algoritmos conocidos: redes neuronales de retropropagación (BP), optimización de enjambre de partículas (PSO) y máquina de vectores de soporte (SVM). El conjunto de datos consta de imágenes de retrodispersión inducidas por láser (es decir, imágenes moteadas de 5472 × 3648 píxeles), donde el proceso de adquisición se lleva a cabo con un sistema láser con un expansor de haz, una cámara a color de semiconductor de óxido de metal complementario (CMOS) con una lente de zoom y un polarizador. El conjunto de datos tiene un total de 500 muestras de manzanas de un tamaño similar (diámetro ecuatorial de 80 a 100 mm). El modelo CNN propuesto para la detección de manzanas logra una tasa de reconocimiento del 92,50%.

En la literatura otros autores han trabajado el problema de detección de daños en manzana y de otros vegetales, se han publicado resultados muy interesantes, a continuación, se mencionan algunos.

Sachin *et al.*, (2019), detectaron vegetales usando el algoritmo YOLO, las imágenes se preprocesaron antes del entrenamiento dibujando cuadros delimitadores alrededor de la verdura manualmente usando la herramienta OpenCV. Este método proporcionó una forma rápida y eficiente para identificar un objeto en la imagen o video dado. Una vez que la red fue entrenada, se realizaron pruebas y la salida mostró cuadros delimitadores alrededor de la verdura reconocida y la etiquetó con su categoría de clase prevista, los autores reportan una precisión del 61,6%.

Sharma et *al*., (2015) trabajaron en reconocimiento de lesiones por antracnosis en superficies de Manzana usando YOLOV 3-Denso, proponen una estrategia de identificación de lesiones por antracnosis dependiente del aprendizaje profundo. El sistema neuronal convolucional (DenseNet) se utiliza para optimizar las capas de extracción de características del modelo YOLO-V3. El modelo mejorado de YOLO-v3-Denso, superó a Faster R-CNN con VGG16-NET, los autores informan que la técnica propuesta se puede aplicar en general al reconocimiento de lesiones por antracnosis en superficies de manzanas incluso en plantaciones.

Raheel Siddiqi (Siddiqi *et al*., 2019) trabajaron en detección automatizada de defectos de manzanas utilizando el estado del arte de técnicas de detección de objetos y proponen dos sistemas diferentes que pueden realizar la tarea de detección de defectos de manzana. Sus propuestas se basan en los modelos de detección de objetos SSD (del inglés “Single Shot Detector”) y YOLOv2. Crearon un conjunto de datos de 244 imágenes de manzanas defectuosas para el entrenamiento y las pruebas de los sistemas de detección. Los resultados de rendimiento son alentadores para ambos sistemas propuestos. Sin embargo, en sus conclusiones informan que el sistema basado en SSD mostró un rendimiento superior al del sistema basado en YOLOv2.

Chen *et al*., (2021), propusieron un sistema para identificar la calidad externa de la fruta, que utiliza una cámara como sensor de imagen y un algoritmo de inteligencia artificial como clasificador. Esta aplicación fue adecuada para entornos operativos reales. Las frutas son detectadas principalmente por el algoritmo YOLOv3. La aplicación propuesta detectó frutas redondas como manzanas, naranjas y limones. También reportan que su sistema cuenta con una interfaz gráfica de usuario para controlar y recopilar datos, evaluar modelos y monitorear todo el funcionamiento del sistema para mejorar la

eficiencia de la aplicación propuesta. Los resultados experimentales muestran que la aplicación propuesta logra una tasa de precisión de hasta el 88 % después de probar 6,000 imágenes de frutas.

Huang *et al.,* (2021), Presentaron un método de detección de manzanas inmaduras basado en YOLOv3 mejorado, la identificación de manzanas inmaduras es un eslabón técnico clave para realizar el monitoreo automático en tiempo real de los huertos, la toma de decisiones experta y la realización de la predicción de producción del huerto. Por lo tanto, proponen un método mejorado de detección de YOLOv3 para manzanas inmaduras en la escena del huerto. Para su conjunto de datos con frutos gravemente ocluidos, la puntuación F1 y la media del promedio de precisión (mAP) del modelo de reconocimiento de manzanas inmaduras propuesto fue 0,652 y 0,675, respectivamente. La velocidad de inferencia para una sola imagen de 416 × 416 es de 12 ms, la velocidad de detección puede alcanzar los 83 fotogramas/s en una GPU 1080ti y la velocidad de inferencia es de 8,6 ms. Por lo tanto, para el conjunto de datos de manzanas inmaduras severamente ocluidas, el método propuesto en este artículo tiene un efecto de detección significativo y proporciona una solución factible para la automatización y mecanización de la industria de la manzana.

La aportación de Hu *et al.*, (2021), es significativa ya que desarrollaron un trabajo para la detección y clasificación de manzanas en el campo, basadas en la fusión de funciones múltiples. Se desarrolló y utilizó un dispositivo para detectar y clasificar manzanas en el campo utilizando un método de aprendizaje profundo. Se seleccionaron cuatro características para clasificar las manzanas, el tamaño, el color, la forma y los defectos de la superficie, y se diseñaron algoritmos de detección para discriminar entre las cuatro características utilizando la visión artificial y otros métodos. Luego, se fusionaron las cuatro características de la manzana y se utilizó una máquina de vectores de soporte (SVM) para clasificar las manzanas en el campo en tres grados: fruta de primera clase, fruta de segunda clase y fruta de otra clase. Los resultados mostraron que, para un solo índice, la precisión para detectar el tamaño de la manzana, la forma de la fruta, el color y los defectos de la superficie fueron del 99.04 %, 97.71 %, 98 % y 95.85 %. Las precisiones de clasificación para la fruta de primera clase, la fruta de segunda clase, la fruta de otra clase y la precisión de clasificación promedio basada en múltiples características fueron 94.55%, 95.71%, 100% y 95.49%, respectivamente. El experimento de campo mostró que la precisión de clasificación promedio fue del 94.12 % cuando el intervalo de alimentación de las manzanas fue inferior a 1.5 s. y la velocidad de marcha (del dispositivo) no superó los 0.5 m/s, cumpliendo con los requisitos de precisión de campo para clasificación de manzanas.

Finalmente, Nguyen *et al*., (2022), realizaron detección temprana de magulladuras leves en manzanas mediante imágenes de infrarrojo cercano, su objetivo fue evaluar la posibilidad de aplicar imágenes NIR para detectar hematomas leves en las manzanas. Concluyen que es prometedor desarrollar dispositivos con estas características para detectar leves magulladuras no solo en manzanas, sino también en otras frutas con piel suave y delgada en sus primeras etapas de daño.

3 MARCO TEÓRICO

3.1 Clasificación con redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son un tipo de algoritmo de aprendizaje profundo que ha demostrado ser muy efectivo en la clasificación de calidad y defectos en los frutos agrícolas. Las CNN pueden identificar patrones y características en las imágenes de frutos agrícolas, lo que permite una detección más precisa de los defectos. Las capas convolucionales de las CNN realizan una serie de operaciones de filtrado que detectan bordes, texturas y otras características importantes en las imágenes.

Una propiedad particular de las redes neuronales convolucionales (CNN), es que en cada capa convolucional extrae patrones locales en pequeñas ventanas de dos dimensiones (kernel) orientadas a detectar características o rasgos visuales como aristas, líneas, texturas, color, entre otras. Además, pueden extraer información de los defectos del fruto en función de las imágenes con las que son entrenadas. Debido a estas cualidades, las CNN fueron consideradas para el desarrollo de esta investigación.

3.1.1 Red convolucional

La primera red neuronal convolucional fue creada por Yann LeCun y estaba enfocada en el reconocimiento de letras manuscritas (LeCun *et al.*, 1998). Esta arquitectura se muestra en la Figura 3.1 y utiliza varias capas para la extracción de características de las imágenes de entrada.

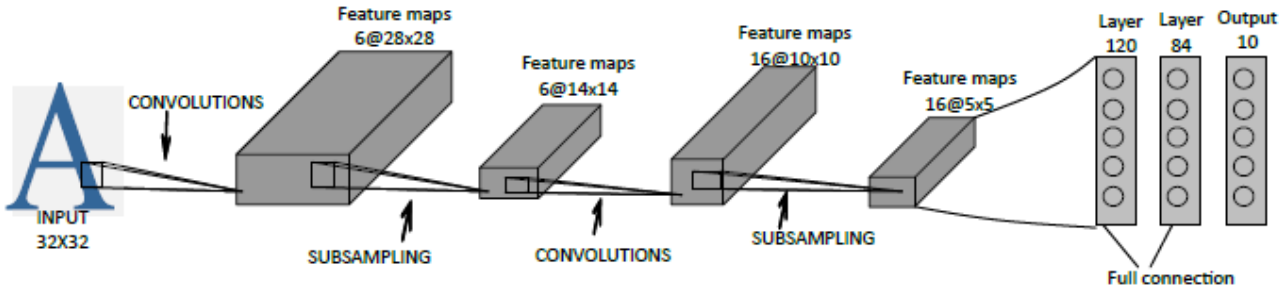


Figura 3.1. Red neuronal convolucional creada por Yann LeCun (LeCun, 1998)

Las capas utilizadas en Lenet5 se pueden organizar en: capa de entrada, capa convolucional, capa de reducción o agrupación (pooling) y una capa totalmente conectada como se observa en la Figura 3.2. En la primera capa; la red toma como valores de entrada los pixeles de las imágenes, la segunda, extrae las características de las imágenes de entrada, en la tercera capa se reduce la cantidad de parámetros, conservando las características más comunes para que la red pueda identificar el objeto de interés.

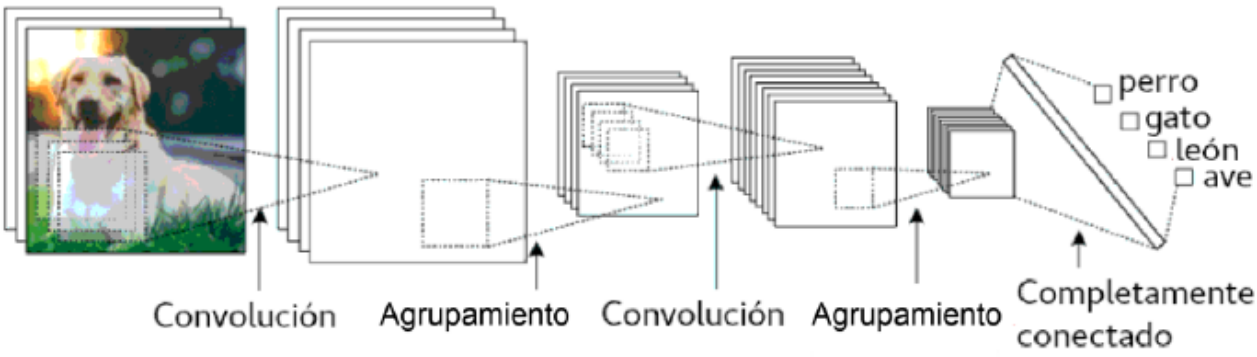


Figura 3.2. Estructura común de una red neuronal convolucional (Programador Clic, 2020)

Las redes neuronales convolucionales son redes multicapa inspiradas en el córtex visual humano, esta arquitectura se utiliza principalmente para el procesamiento de imágenes. Una convolución es una operación lineal especial que se utiliza en procesamiento de imágenes. Las capas convolucionales son responsables de realizar esta operación, donde se toman "grupos de píxeles cercanos" de la imagen de entrada y se realiza un producto escalar con una pequeña matriz llamada kernel o núcleo.

Matemáticamente, una convolución se define como el producto de dos funciones que contienen números reales como argumentos, y la salida dicha operación es también otra función.

Para explicar la operación de convolución, se define lo siguiente:

Sea f y g dos funciones, en donde la operación de convolución $(f * g)$ de estas funciones, está representada por la ecuación 1:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau \qquad (1)$$

Cuando se trabaja con imágenes, como es el caso de este trabajo de investigación, la información que proviene de estas son datos discretos, por lo tanto, la integral definida en la ecuación 1, se convierte en una sumatoria de funciones continuas, tal y como se muestra en la ecuación 2:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m] \qquad (2)$$

En las redes neuronales convolucionales (CNN), el primer término de la operación de convolución es la imagen para procesar, $x[n]$, mientras que el segundo termino se refiere al kernel con el que se procesa, $h[n]$. Para una kernel con dominio finito, la convolución está definida en el dominio $\{0,1,\dots,k-1\}$.

La convolución realiza para cada valor de la imagen K multiplicaciones y $K-1$ sumas, como se muestra en la ecuación 3:

$$(x * h)[n] = \sum_{K=0}^{K-1} h[k]x[n-k] \qquad (3)$$

En el caso multidimensional, las variables $[n1, n2]$, corresponden a los pixeles de la imagen y si se considera que los kernels que se utilizan tienen tamaños k_1 y k_2, entonces la convolución se define como la ecuación 4.

$$(x * h)[n1, n2] = \sum_{k_1=0}^{k_1-1} \sum_{k_2=0}^{k_2-1} h[k_1, k_2]x[n1 + k_1\,, n2 + k_2] \qquad (4)$$

3.1.2 Características del Núcleo (kernel) en una CNN

El kernel, también conocido como filtro, es una matriz de pesos que se aplica a cada sección de la imagen de entrada para extraer características relevantes. Durante la convolución, el kernel se desliza sobre la imagen de entrada y realiza operaciones matemáticas para generar un mapa de características. Este mapa de características resalta las partes importantes de la imagen, como los bordes, los patrones y los objetos.

El tamaño y la forma del kernel pueden variar según la tarea de la red neuronal convolucional (CNN). Por ejemplo, un kernel más grande puede ser de utilidad para detectar características más grandes en la imagen, mientras que un kernel más pequeño puede detectar características más finas.

Durante la convolución, el kernel se desplaza de izquierda a derecha por cada fila de la imagen de entrada hasta llegar a la esquina inferior derecha. Este proceso se repite iterativamente hasta cubrir completamente la superficie de la imagen. La forma en que se realiza la convolución se define mediante dos parámetros: el stride, que determina el desplazamiento entre cada movimiento del kernel, y el padding, que consiste en agregar ceros alrededor de los valores de la imagen de entrada.

Cada filtro o kernel tiene como objetivo detectar una característica particular en cada ubicación de entrada, por lo tanto, la traducción espacial de la entrada desde una capa de detección de características se transferirá a la salida sin cambios (LeCun *et al*., 1998). En el proceso de convolución, un pequeño filtro deslizante opera de izquierda a derecha a través de la imagen de arriba a abajo. En la Figura 3.3, se muestra un ejemplo de la operación de convolución con una imagen de entrada (4 × 4) y un núcleo de convolución (3 × 3), que da como resultado una imagen de salida convolucionada. En cada ubicación, se calcula la suma de los productos entre cada elemento del núcleo y el elemento de entrada correspondiente. Este proceso se repite utilizando diferentes núcleos para formar tantos mapas de características de salida como se desee.

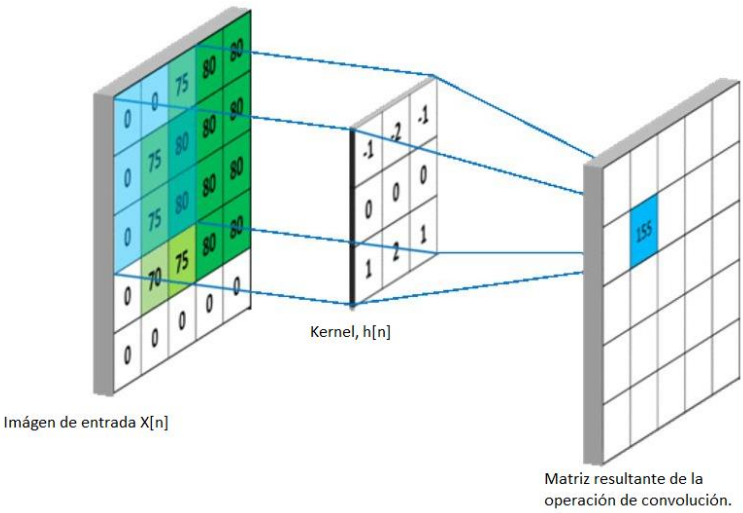


Figura 3.3. Muestra un ejemplo de la operación de convolución con una imagen de entrada (4 × 4) y un núcleo de convolución (3 × 3) (Programador Clic, 2020).

Los filtros (kernels) son elementos claves para extraer características de la imagen, por ejemplo, el kernel laplaciano permite detectar regiones similares a bordes mediante el operador laplaciano (Figura 3.4, "filtro Laplacian"). El kernel gaussiano es una matriz bidimensional que se utiliza para suavizar imágenes mediante una convolución que sigue la distribución gaussiana, proporcionando un efecto de suavizado y eliminación de ruido controlado (Figura 3.4, "filtro GaussianBlur"), mientras que el Kernel "blur", permite realizar un desenfoque (Figura 3.4, "filtro blur").

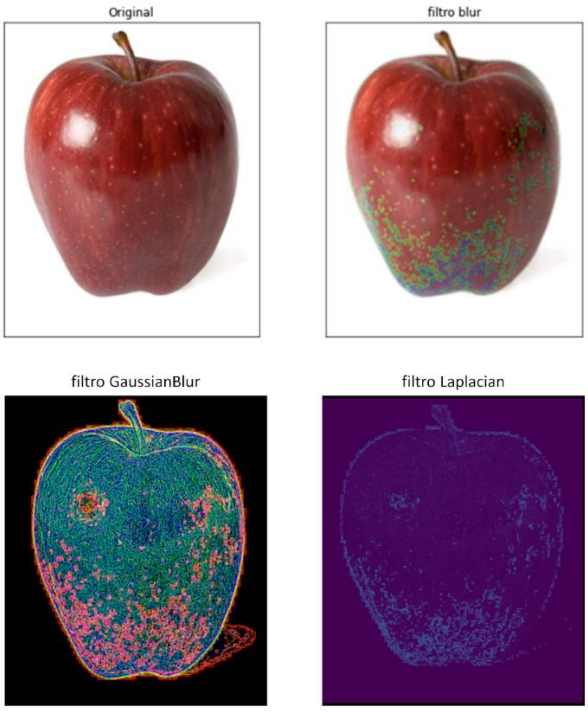


Figura 3.4. Muestra el resultado de aplicar a una imagen operaciones de convolución con distintos nucleos (kernel)

3.1.3 Capa completamente conectada

Capa totalmente conectada (FC): la salida final de las etapas convolucionales se aplana en una matriz 1D y se conecta a una capa totalmente conectada (Figura 3.5). Las capas FC toman los resultados del proceso de convolución/agrupación y los usan para clasificar la imagen en una etiqueta (es decir, clase), como una red neuronal tradicional. Por lo tanto, la función de activación de la última capa (la capa de salida) calcula las probabilidades finales de cada clase y se selecciona de acuerdo con la tarea. Por lo general, una tarea de clasificación de varias clases utiliza la función Softmax, donde el valor de probabilidad de cada clase oscila entre [0, 1] y su suma total es igual a 1. Finalmente, cada neurona de salida decide sobre cada una de las etiquetas, y el mayor valor de salida corresponde a la decisión de clasificación.

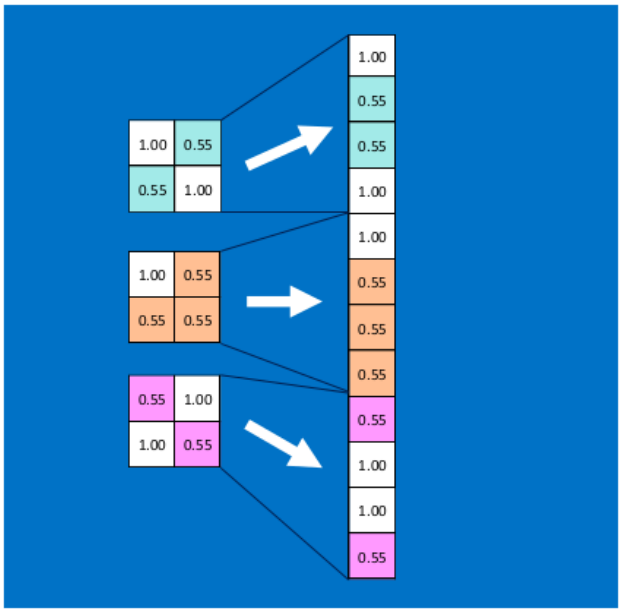


Figura 3.5. Aplanado de las etapas convolucionales en una matriz 1D (Programador Clic, 2020)

3.1.4 Consideraciones para el entrenamiento de una CNN

En el proceso de entrenamiento se optimizan diferentes parámetros de capa de la red neuronal para minimizar las diferencias entre las etiquetas dadas en un conjunto de datos de entrenamiento y las predicciones de salida. Comúnmente, el algoritmo de retropropagación es el método más utilizado para entrenar redes neuronales convolucionales.

Para poder entrenar una CNN, se deben tomar en cuenta los siguientes aspectos:

• Definir la arquitectura CNN: esto consiste en establecer el número de capas, así como el tamaño y número de filtros para cada capa. El diseño de la arquitectura siempre dependerá del objetivo de la CNN.
• Función de pérdida: estas miden la diferencia entre las etiquetas verdaderas de y las salidas de la red (predicción).

Por lo general, se aplica la función de error cuadrático medio y viene dada por la ecuación (1):

$$L = \sum (Etiqueta\ verdadera - Salida)^2 \quad (1)$$

Por lo tanto, L debe minimizarse para encontrar la contribución de cada peso y optimizarlos. El algoritmo de descenso de gradiente se adopta ampliamente para el procedimiento de minimización, que se expresa matemáticamente como derivada parcial de la función de pérdida. Entonces, el proceso de actualización de parámetros se formula de la siguiente manera y se presenta en la ecuación (2):

$$W_k = W_{k-1} - \alpha * \frac{\partial L}{\partial W} \quad (2)$$

Donde α denota la tasa de aprendizaje. Por lo tanto, la tasa de aprendizaje es un hiperparámetro muy importante y debe establecerse antes de iniciar el proceso de entrenamiento. Es importante señalar que una tasa de aprendizaje muy baja puede dar un resultado más preciso, pero la red puede tardar más en entrenarse.

• Conjunto de datos de entrenamiento: los datos disponibles generalmente se dividen en tres subconjuntos: un conjunto de entrenamiento para entrenar la red, el conjunto de validación para evaluar el modelo durante el proceso de entrenamiento y el conjunto de prueba para evaluar el modelo entrenado final. La mayoría de las arquitecturas CNN requieren que todos los datos de entrenamiento tengan la misma forma (es decir, dimensiones). Por lo tanto, el preprocesamiento de los datos es el primer paso antes del proceso de entrenamiento para normalizar los datos.

Otro punto importante es que el conjunto de datos debe estar equilibrado (balanceado), lo que significa la misma cantidad de imágenes para cada clase. En caso de que el conjunto de datos no tenga un número suficiente de imágenes, se recomienda aplicar una técnica de aumento de datos (“data argumentation”). Esta consiste en aumentar la cantidad de datos de entrenamiento realizando a las imágenes una serie de transformaciones, como rotaciones, traslaciones, espejos, entre otras.

Tomando en cuenta estas consideraciones. El algoritmo para el entrenamiento de una CNN es el siguiente:

1. Seleccione un conjunto de datos de imágenes de entrenamiento, generalmente tomadas por lotes con dimensiones menores en RGB (3 canales) o escala de grises (1 canal) y se les aplica normalización.
2. Pase cada lote por la red y obtenga el resultado.
3. Calcule el error entre las etiquetas dadas y las predicciones de salida utilizando una función de pérdida L (ecuación 1).
4. Propague el error por toda la red mediante el algoritmo de retropropagación.
5. Actualice los pesos W para minimizar el error.
6. Repetir hasta converger o alcanzar un límite de iteraciones.

3.1.5 Métricas del rendimiento en clasificación

La matriz de confusión es una herramienta fundamental para evaluar el rendimiento de algoritmos de clasificación supervisada. Su objetivo es mostrar la distribución de los valores reales y las predicciones realizadas por el algoritmo. Esta matriz proporciona una representación clara de las confusiones entre clases, donde cada columna representa el número de predicciones para una clase específica y cada fila representa las instancias pertenecientes a la clase real. Dentro de la matriz de confusión la diagonal principal representa las predicciones verdaderas, es decir, para una clase en específico los valores que se encuentren dentro de esta diagonal, representan a los verdaderos positivos, mientras que los datos que se encuentran sobre esa diagonal, pero no pertenezcan a esta clase serán definidos como verdaderos negativos, por lo tanto, será positivo si se encuentra en la clase de interés y negativo si se encuentra en cualquier otra clase.

Las evaluaciones incorrectas se refieren a los falsos positivos y falsos negativos que se encuentren dentro de la matriz. Los falsos positivos se refieren a las predicciones erróneas sumándose a la clase incorrecta y los falsos negativos son aquellos valores reales de la clase de interés que fueron mal clasificados, los cuáles serán restados de esta. A partir de estos valores pueden ser calculados algunos parámetros de desempeño como; precisión (precision), sensibilidad (recall), puntuación F1 (F1-score) y exactitud (accuracy) que pueden ser empleados para comparar el desempeño entre redes, o bien, para evaluar si la red tiene un buen o mal comportamiento en términos de clasificación.

La precisión se define como la medida de cercanía entre el resultado de una predicción y el valor verdadero. Se calcula como el cociente entre los datos correctamente clasificados y el total de predicciones positivas, ver ecuación 3.

$$Precisión = \frac{TP}{TP + FP} \qquad (3)$$

La sensibilidad, que también se le conoce como tasa de verdaderos positivos, se define como la proporción de casos positivos que son identificados de manera correcta por el modelo y se calcula utilizando la ecuación 4:

$$Sensibilidad = \frac{TP}{TP + FN} \qquad (4)$$

La puntuación F1, es empleada de manera que combina las medidas de precisión y sensibilidad, esto hace que sea más practico comparar el rendimiento, siendo de gran utilidad principalmente cuando se tiene una distribución de clases desigual y se calcula utilizando la ecuación 5.

$$Puntuación\ F1 = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad} \qquad (5)$$

En cuanto al valor de la exactitud, éste se refiere a la medida de cercanía entre el resultado de una medición y el valor verdadero. Se define como el porcentaje de predicciones correctas en comparación

con el total de casos. Es el cociente entre los casos correctamente clasificados por el modelo y la suma de todos los casos y se calcula utilizando la ecuación 6.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \qquad (6)$$

Para las ecuaciones 1, 2 y 4, el término TP = Verdaderos positivos, TN = Verdaderos negativos, FP = Falsos positivos y FN = Falsos negativos.

3.2 Detección con redes neuronales convolucionales

Clasificar y localizar objetos en una imagen se denomina detección de objetos. Un enfoque común es utilizar una red neuronal convolucional (CNN) entrenada para clasificar y deslizarla por la imagen para ubicar un objeto.

Como se muestra en la Figura 3.6, la imagen está en una cuadrícula de 14 × 14 y una CNN (representada por el rectángulo negro grueso) se desliza por todas las regiones de 5 × 5. Cuando la CNN está analizando la parte superior izquierda de la imagen, detecta parte de la manzana que se encuentra más a la izquierda, y luego detecta esa misma manzana nuevamente cuando se desplaza por primera vez un paso hacia la derecha ("stride = 1"). En el siguiente paso, comienza a detectar parte de la manzana que se encuentra más a la derecha, y luego vuelve a detectar esta misma manzana una vez que se mueve un paso más hacia la derecha. Así sucesivamente se continúa deslizando la CNN a través de toda la imagen, observando todas las regiones de 5 × 5.

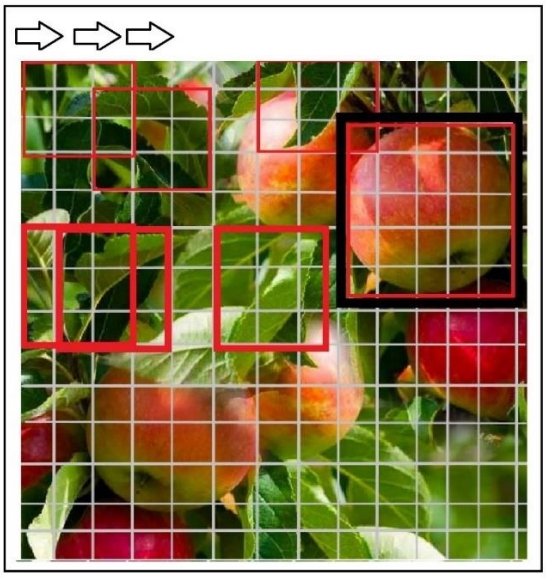


Figura 3.6. Muestra un ejemplo de ventana deslizante en una imagen que contiene manzanas.

Dado que los objetos pueden tener diferentes tamaños, también se deslizaría la CNN en la imagen a través de regiones de diferentes tamaños. Por ejemplo, una vez que se haya terminado regiones de tamaño 5× 5, es posible deslizar la CNN en todas las regiones de 3 x 3.

En general, se trata de una técnica sencilla, pero como se puede ver en la imagen de la Figura 3.6, una de las principales desventajas es que detectará el mismo objeto varias veces, en posiciones ligeramente diferentes. Luego entonces, es necesario un procesamiento posterior para deshacerse de todos los cuadros delimitadores innecesarios. Este enfoque simple para la detección de objetos funciona relativamente bien, pero una de sus principales limitaciones es que requiere ejecutar la CNN muchas veces, por lo que el proceso se vuelve bastante lento. Afortunadamente, hay una forma mucho más rápida de deslizar una CNN a través de una imagen, y es utilizando una red totalmente convolucional.

3.2.1 Algoritmo de detección YOLO

YOLO ("You Only Look Once") pertenece a una familia de algoritmos que aprovecha el uso de redes neuronales totalmente convolucionales (FCN) para la detección de objetos. Es uno de los algoritmos de detección de objetos más rápidos disponibles y una buena opción para la detección en tiempo real. YOLO es una arquitectura propuesta por Joseph Redmon en 2016 (Redmon J. *et al.*, 2016), y mejorado con la segunda versión en 2017 (Redmon, J. *et al.*, 2017) (YOLOv2) y con una tercera versión en 2018 en donde también participa como autor (Redmon *et al.*, 2018) (YOLOv3).

YOLO es una red neuronal convolucional que realiza predicciones simultáneas de cuadros delimitadores y probabilidades de clase para varios objetos en una imagen. La imagen se divide en una cuadrícula de tamaño S x S, donde cada celda de la cuadrícula es responsable de detectar objetos si se encuentran en esa celda, esto se puede observar con mayor detalle en la Figura 3.7.

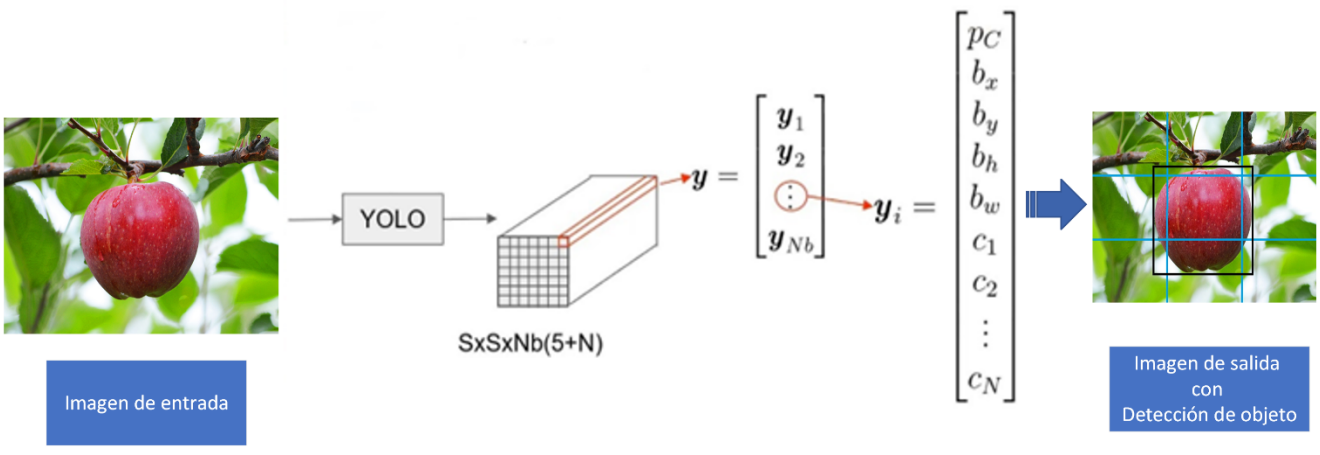


Figura 3.7. Muestra la capa de salida como una cuadricula de tamaño S x S, con la que YOLO realiza la detección de objetos.

La capa de salida de YOLO es la capa final en la arquitectura de la red neuronal que se encarga de generar las predicciones de detección de objetos. Esta capa de salida se compone de varias celdas de la malla de tamaño S x S, cada una de las cuales se encarga de predecir la presencia de objetos en una subregión específica de la imagen. También, cada celda de la malla se divide en varios cuadros delimitadores ("bounding box") que representan las diferentes posibilidades de localización de objetos en esa subregión.

Esto implica que cada cuadro delimitador ("bounding box") es asociado con un conjunto de características, como las coordenadas de la caja, la clase del objeto y la confianza en la detección (ver Figura 3.7). Las coordenadas de la caja se refieren a las esquinas superior izquierda e inferior derecha del cuadro delimitador y la clase del objeto se refiere a la categoría a la que pertenece el objeto detectado (C1, C2, …, Cn). La confianza es un valor entre 0 y 1 que indica la probabilidad de que el objeto detectado sea real.

La confianza se calcula utilizando una función de pérdida en la capa de salida de YOLO, que compara las predicciones del modelo con las etiquetas de entrenamiento verdaderas. La función de pérdida también tiene en cuenta la probabilidad de que un objeto esté presente en una celda de la malla, así como la probabilidad de que un objeto pertenezca a una determinada clase.

La confianza es un factor importante porque ayuda a eliminar detecciones falsas o imprecisas. Es decir, si un modelo entrenado tiene una baja confianza en una detección, es probable que sea incorrecto, y con esto se puede descartar. Por otro lado, si un modelo entrenado tiene una alta confianza en una detección, es probable que sea correcto y se puede utilizar para tomar decisiones.

La capa de salida de la arquitectura YOLO genera una serie de predicciones para cada objeto detectado en la imagen, junto con información de localización precisa y confianza en la detección. Estas predicciones se utilizan para dibujar las cajas delimitadoras alrededor de los objetos detectados en la imagen.

La arquitectura YOLO considera la detección de objetos como un problema de regresión. Predice directamente las probabilidades de clase y las compensaciones del cuadro delimitador a partir de imágenes completas con una única red neuronal totalmente convolucional.

Los cuadros delimitadores están compuestos de 4 valores: x, y, w, h. Las coordenadas (x, y) representan el centro del cuadro en relación con los límites de la celda de la cuadrícula. Como se puede observar en la Figura 3.8, el punto central está marcado en color negro con coordenadas (290,240).

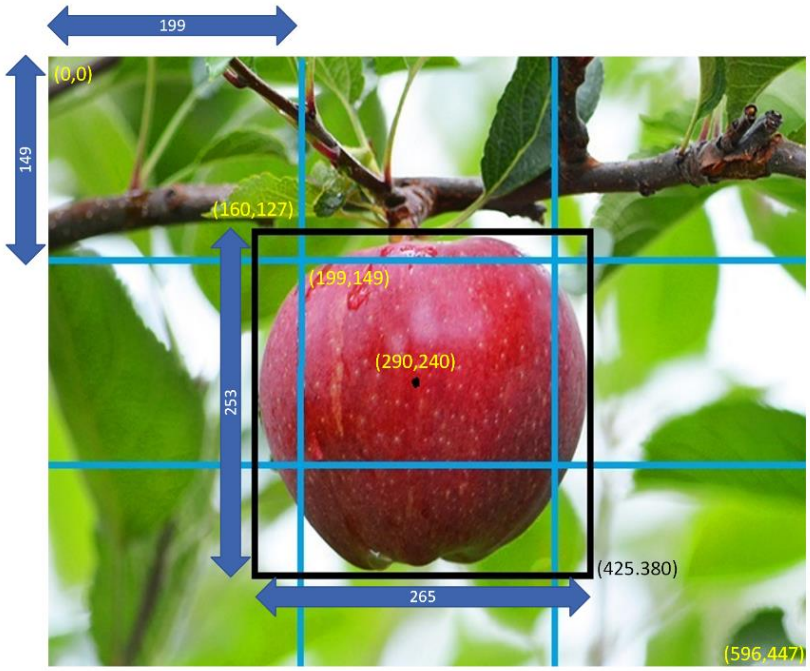


Figura 3.8. Ejemplo de las coordenadas normalizadas del cuadro de una imagen.

Para calcular estos valores del punto central en referencia con la celda, se debe considerar lo siguiente. Si las celdas tienen un tamaño de (199 x149), como el caso que estamos analizando (Figura 3.8), se calcula la normalización respecto a la celda, de la siguiente manera: x = (290-199) / 199 = 0.45, además, y = (240-149) / 149 = 0.61.

La anchura (w) y la altura (h) se toman en relación con la imagen completa, es decir, que, a diferencia de la normalización del punto central, en este caso la normalización se toma en relación con el tamaño de la imagen. Lo que se hace es tomar el ancho de la detección del objeto (cuadro de color negro en la Figura 3.8), y la altura, y se normaliza de la siguiente manera: w = 265/597 = 0.44, también, h = 253/448 = 0.56. Considerando que el tamaño de la imagen en este caso es de (597x448).

3.2.2 Puntuación de confianza

El método YOLO divide la imagen de entrada en pequeñas celdas de cuadrícula S×S. Si el centro de un objeto cae en una celda de la cuadrícula, la celda es responsable de detectar al objeto. Cada celda de la cuadrícula predice la información de posición de los cuadros delimitadores B (Nb, ver Fig. 16) y calcula las puntuaciones de confianza (PC) correspondientes a estos cuadros delimitadores. Cada puntaje de confianza se puede obtener de la siguiente manera, como se muestra en la ecuación 3:

$$C_i^j = P_{i,j}(Objeto) * IoU_{C_{predicho}}^{C_{verdadero}} \qquad (3)$$

donde C_i^j es la puntuación de confianza del j-ésimo cuadro delimitador en la i-ésima celda de la cuadrícula. $P_{i,j}(Objeto)$ es simplemente una función de presencia o ausencia de un objeto. El $IoU_{C_{predicho}}^{C_{verdadero}}$ representa la intersección sobre la unión (IOU) entre el cuadro predicho y el cuadro verdadero. En otras palabras, si no hay un objeto en esa celda, las puntuaciones de confianza deben ser cero.

El método YOLO utiliza la entropía cruzada binaria de las puntuaciones de confianza predichas y las puntuaciones de confianza verdadera como una parte de la función de pérdida J (ecuación 4) y se puede expresar de la siguiente manera:

$$J = \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\hat{C}_i^j \log\left(C_i^j\right) - (1 - \hat{C}_i^j) \log\left(1 - C_i^j\right) \right] \quad (4)$$

Donde S^2es el número de celdas de la cuadrícula de la imagen y B es el número de cuadros delimitadores. C_i^j y $\hat{C}_i^j$ son la puntuación de confianza predicha y la puntuación de confianza verdadera, respectivamente.

3.2.3 Predicción de cuadros delimitadores.

La posición de cada cuadro delimitador se basa en cuatro predicciones: t_x, t_y, t_w, t_h, suponiendo que (c_x, c_y) es el desplazamiento de la celda de la cuadrícula desde la esquina superior izquierda de la imagen. La posición central de los cuadros delimitadores predichos finales está desplazada desde la esquina superior izquierda de la imagen dada por (b_x, b_y) y se calculan como se muestra en las ecuaciones $5a$ y $5b$.

$$b_x = \sigma(t_x) + c_x \quad (5a)$$

$$b_y = \sigma\left(t_y\right) + c_y \quad (5b)$$

donde $\sigma()$ es una función sigmoidea. El ancho y la altura del cuadro delimitador predicho se calculan como se muestra en las ecuaciones $6a$ y $6b$.

$$b_w = P_w e^{t_w} \quad (6a)$$

$$b_h = P_h e^{t_h} \quad (6b)$$

donde P_w, P_h son el ancho y el alto del cuadro delimitador anterior.

El cuadro delimitador verdadero consta de cuatro parámetros (gx, gy, gw y gh), que corresponden a los parámetros predichos bx, by, tw y th, respectivamente. Con base en $(5a)$, $(5b)$, $(6a)$ y $(6b)$, los valores verdaderos de $\hat{t}_x$, $\hat{t}_y$, $\hat{t}_w$ y $\hat{t}_h$, se pueden obtener de la siguiente manera:

$$\sigma(\hat{t}_x) = g_x - c_x \quad (7a)$$

$$\sigma(\hat{t}_y) = g_y - c_y \quad (7b)$$

$$\hat{t}_w = \log(g_w/p_w) \quad (8a)$$

$$\hat{t}_h = \log(g_h/p_h) \quad (8b)$$

3.2.4 Error cuadrático en la predicción de coordenadas

El método YOLO utiliza el error cuadrático de la predicción de coordenadas como una parte de la función de pérdida, la cual se puede expresar de la siguiente manera:

$$J_2 = \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left(\sigma(t_x)_i^j - \sigma(\hat{t}_x)_i^j\right)^2 + \left(\sigma(t_y)_i^j - \sigma(\hat{t}_y)_i^j\right)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left(\sigma(t_w)_i^j - \sigma(\hat{t}_w)_i^j\right)^2 + \left(\sigma(t_h)_i^j - \sigma(\hat{t}_h)_i^j\right)^2 \right] \quad (9)$$

3.2.5 El cálculo de la intersección sobre la unión (IoU)

El cálculo de la intersección sobre la unión (IoU) se utiliza para medir la superposición o similitud entre dos regiones delimitadas, como, por ejemplo, los cuadros delimitadores de objetos detectados en una imagen. La intersección sobre la unión se calcula dividiendo el área de intersección entre el área de unión de las dos regiones (ver Figura 3.9). Para obtener la intersección, se determina el área común entre las dos regiones. Luego, se calcula el área de unión sumando el área de cada región y restando el área de intersección, para evitar contar el área común dos veces. El resultado de la IoU es un valor entre 0 y 1. Un valor de 0 indica que no hay superposición entre las regiones, mientras que un valor de 1 indica una superposición completa.

La intersección sobre la unión (IoU) se utiliza generalmente como una métrica para evaluar la precisión de las detecciones de objetos en algoritmos de visión por computadora y aprendizaje automático. Un valor alto de IoU indica una detección precisa de los objetos, mientras que un valor bajo puede indicar una detección incorrecta o imprecisa.

Por otra parte, como se explicó en la sección 3.2.1, cada celda de la rejilla predice B cuadros delimitadores donde B es el número de cuadros que se desean predecir en cada celda. Para cada cuadro delimitador, se asigna una puntuación de confianza que refleja la probabilidad de que ese cuadro contenga un objeto.

La puntuación de confianza se calcula considerando múltiples factores, como la probabilidad de clase del objeto y la precisión de la predicción del cuadro delimitador. Esta puntuación se obtiene combinando diferentes salidas de la red neuronal, que pueden incluir valores de probabilidad y otros parámetros aprendidos durante el entrenamiento. Formalmente la confianza se define como P_i (Objeto)*IoU (definido en ecuación 3).

En resumen, la puntuación de confianza de la detección se obtiene a partir del proceso de predicción de cuadros delimitadores y la asignación de una puntuación que refleje la probabilidad y precisión de que ese cuadro contenga un objeto en particular.

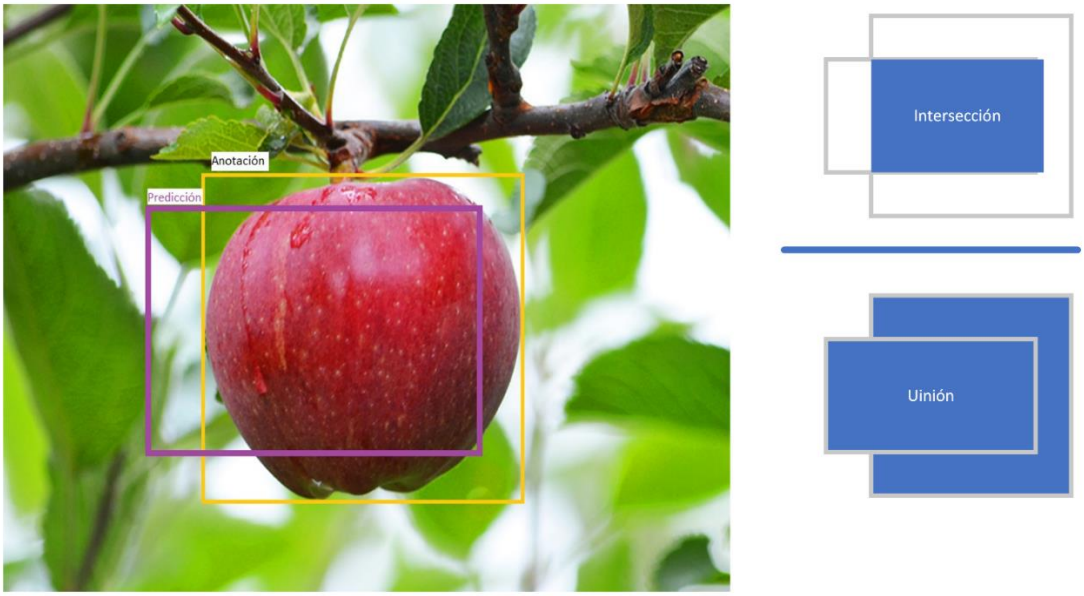


Figura 3.9. Descripción gráfica de la intersección sobre la unión (IoU).

3.2.6 Métrica de la media de la precisión promedio (mAP)

Una métrica utilizada en las tareas de detección de objetos es la media de laprecisión promedio (mAP). Esta métrica de precisión (mAP) se obtiene tras calcular la métrica de precisión promedio (AP). La precisión promedio evalúa el rendimiento de un modelo de detección, calculando la precisión máxima que puede obtener el modelo con al menos 0 % de sensibilidad (“recall”), luego 10 % de sensibilidad (“recall”), 20 %, y así sucesivamente, hasta el 100 % de sensibilidad (“recall”) y luego se calcula la media de estas máximas precisiones. También, cuando hay más de 2 clases, se puede calcular la métrica de precisión promedio (AP) para cada clase y finalmente calcular la media, a esto se le conoce como media de la precisión promedio (mAP).

Sin embargo, un punto importante a considerar es el siguiente, en los sistemas de detección de objetos, hay un nivel adicional de complejidad, ya que el sistema puede detectar una clase correcta, pero en una ubicación incorrecta, y el sistema no debería considerar esta posibilidad como una predicción positiva (es decir, el cuadro delimitador debe estar completamente apagado). Para atender esta situación, un enfoque que se utiliza es definir un valor de umbral para la intersección sobre la unión (IOU): por ejemplo, se puede considerar que una predicción es correcta, solo si el valor de IOU es mayor que, digamos, 0.5, y la clase pronosticada es correcta. En este ejemplo, el mAP correspondiente se indicaría como mAP@0.5 (o mAP@50%, o incluso AP50).

3.2.7 Arquitectura YOLOv1

YOLO es una arquitectura propuesta por Joseph Redmon en 2016 (Redmon J. *et al.*, 2016). La arquitectura YOLO v1 es un enfoque de detección de objetos en tiempo real que se basa en una red neuronal convolucional. Consiste en múltiples capas de convolución y agrupamiento para extraer características de la imagen, seguidas de capas totalmente conectadas que predicen los cuadros delimitadores y las probabilidades de clase. En lugar de utilizar regiones propuestas, YOLO divide la imagen en una cuadrícula y cada celda de la cuadrícula es responsable de detectar objetos. La arquitectura realiza predicciones en una sola pasada, lo que la hace eficiente y rápida. Utiliza una función de pérdida para guiar el aprendizaje durante el entrenamiento. YOLO v1 ha demostrado ser efectiva en aplicaciones de detección de objetos en tiempo real.

La arquitectura de esta red consta de 24 capas convolucionales y 2 capas totalmente conectadas. Los creadores emplean convoluciones 1x1 para disminuir la profundidad de los mapas de características, seguidas de convoluciones 3x3. Estas capas 1x1 y 3x3 se alternan, como se muestra en la Figura 3.10. La última capa convolucional produce un tensor de tamaño (7,7,1024). Luego, el tensor se reduce mediante 2 capas totalmente conectadas, y se obtiene un tensor de salida de tamaño 7x7x30, tal como se observa en la parte final de la Figura 3.10.

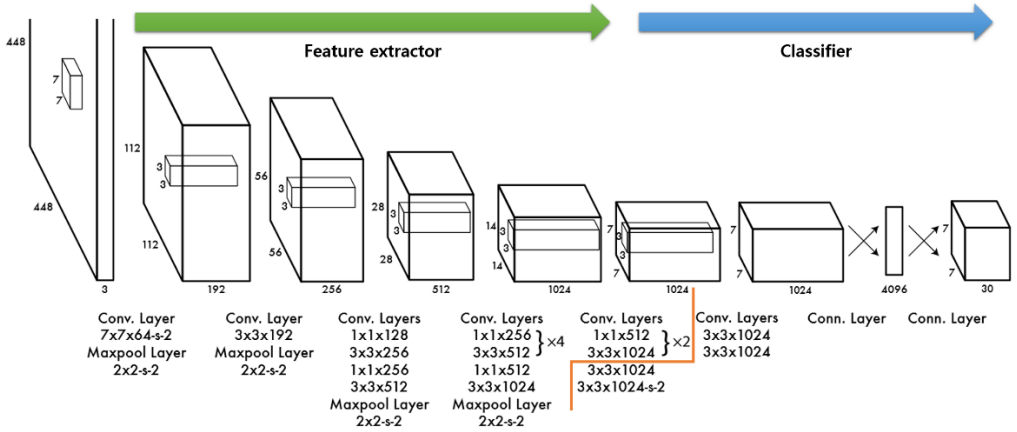


Figura 3.10. Esquema de la arquitectura de la red neuronal YOLOv1 (Redmon J. *et al.*, 2016)

La arquitectura YOLOv1 utiliza la función de activación ReLU (Rectified Linear Unit) en todas sus capas convolucionales, excepto en la última capa convolucional que utiliza la función de activación lineal. La función de activación ReLU es una función no lineal que mapea los valores negativos a cero y mantiene los valores positivos sin cambios. Por otro lado, la función de activación lineal no introduce no linealidades y mantiene los valores sin alterar.

La función de pérdida utilizada en YOLOv1 se conoce como la función de pérdida de detección de objetos basada en la regresión y la clasificación de cuadros delimitadores (del inglés "bounding boxes") y se compone de tres componentes principales: pérdida de coordenadas (del inglés "bounding box regression los"), pérdida de confianza (del inglés "confidence los") y pérdida de clasificación (del inglés "classification los").

3.2.8 Arquitectura YOLOv2

La arquitectura YOLOv2 es un modelo de detección de objetos que fue desarrollado por Redmon J. *et al.*, (2017), es una mejora significativa con respecto a la versión original de YOLOv1.

A continuación, se mencionan algunas de las principales contribuciones de YOLOv2:

- Extracción de características: YOLOv2 utiliza una red neuronal más profunda, con 23 capas convolucionales, la red se basa en la arquitectura Darknet-19, que está diseñada específicamente para la detección de objetos en imágenes. Esta red más profunda permite extraer características más complejas de la imagen, lo que mejora la precisión de la detección de objetos.
- Normalización por lote: En YOLOv2 se utiliza normalización por lote ("batch normalization") en todas las capas convolucionales. Esto ayuda a estandarizar los valores de entrada de las capas y acelerar el entrenamiento.
- Entrenamiento multi-escala: YOLOv2 entrena la red con imágenes de diferentes tamaños. Esto ayuda a que la red aprenda a detectar objetos de diferentes tamaños en una imagen.
- Cuadros de anclaje: YOLOv2 introduce por primera vez, el concepto de anclas ("anchor boxes"), que son cuadros delimitadores predefinidos de diferentes tamaños.

3.2.9 Arquitectura YOLOv3

YOLOv3 es contribución de Redmon J. *et al.*, (2018), y consta de seis tipos de capas, definidas como: red (“net”), convolucional, agrupación (“maxpooling”), yolo, concatenación (“route”) y muestreo ascendente (“upsampling”). La capa de red configura los parámetros de toda la red. La capa de convolución tiene tres módulos: operación de convolución, normalización por lotes y función de activación. Además, la operación de convolución incluye la conversión del mapa de características y la multiplicación general de matrices.

El aprendizaje de características se realiza a través de capas convolucionales, que tiene una estructura similar a ResNet, pero en YOLOv3 se llaman “bloques residuales” y se utilizan para el aprendizaje de características. Los bloques residuales constan de varias capas convolucionales y conexiones de salto. Una característica de YOLOv3 es que realiza la detección en tres escalas diferentes. Las escalas son 13x13, 26x26, y 52x52. La profundidad para la capa yolo se calcula con la siguiente operación:

$$profundidad_{yolo} = (5 + n_{clases}) * n_{anclas} \qquad (10)$$

El algoritmo YOLOv3 requiere, además, que las imágenes de entrada sean múltiplos de 32. Para observar la arquitectura YOLOv3 considerando como imágenes de entrada las imágenes cuya dimensión es 416x416x3 (3, representa los canales RGB y 416x416 es múltiplo de 32) se muestra la Figura 3.11.

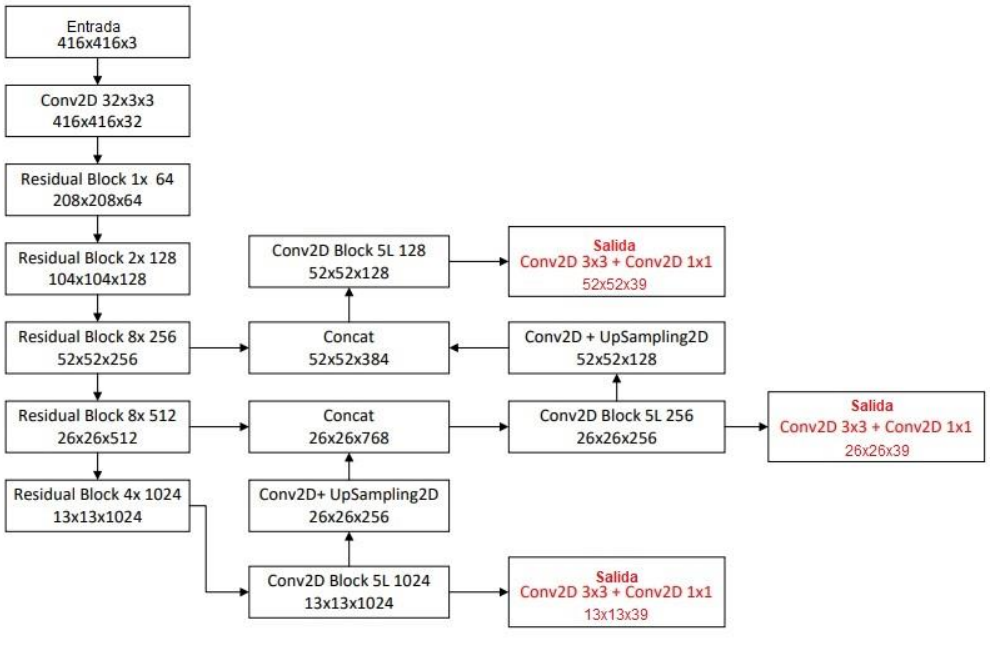


Figura 3.11. Muestra la estructura a bloques de la arquitectura YOLOv3.

3.2.10 Arquitectura YOLOv3-Tiny

Existe otra versión de YOLOv3, la cual tiene una profundidad reducida de la capa convolucional y se llama YOLOv3-Tiny. Una característica de esta arquitectura es que tiene solo capas convolucionales para la extracción de características, lo cual ocasiona que la velocidad de ejecución aumente significativamente. YOLO v3-Tiny usa una capa de agrupación ("max-pooling") y con esto reduce la imagen en la capa de convolución. Predice mediante un tensor tridimensional que contiene puntuación de confianza, cuadro delimitador y predicciones de clase en dos escalas diferentes. Divide una imagen en celdas de cuadrícula S×S. Para las detecciones finales, ignora los cuadros delimitadores para los que la puntuación de confianza no es la mejor utilizando el método de supresión de no máximos (NMS).

La predicción de los cuadros delimitadores se produce en dos escalas de mapa de características, que son 13×13 y 26×26. Para observar la arquitectura YOLOv3-tiny para las imágenes de entrada con dimensión 416x416x3 se muestra la Figura 3.12.

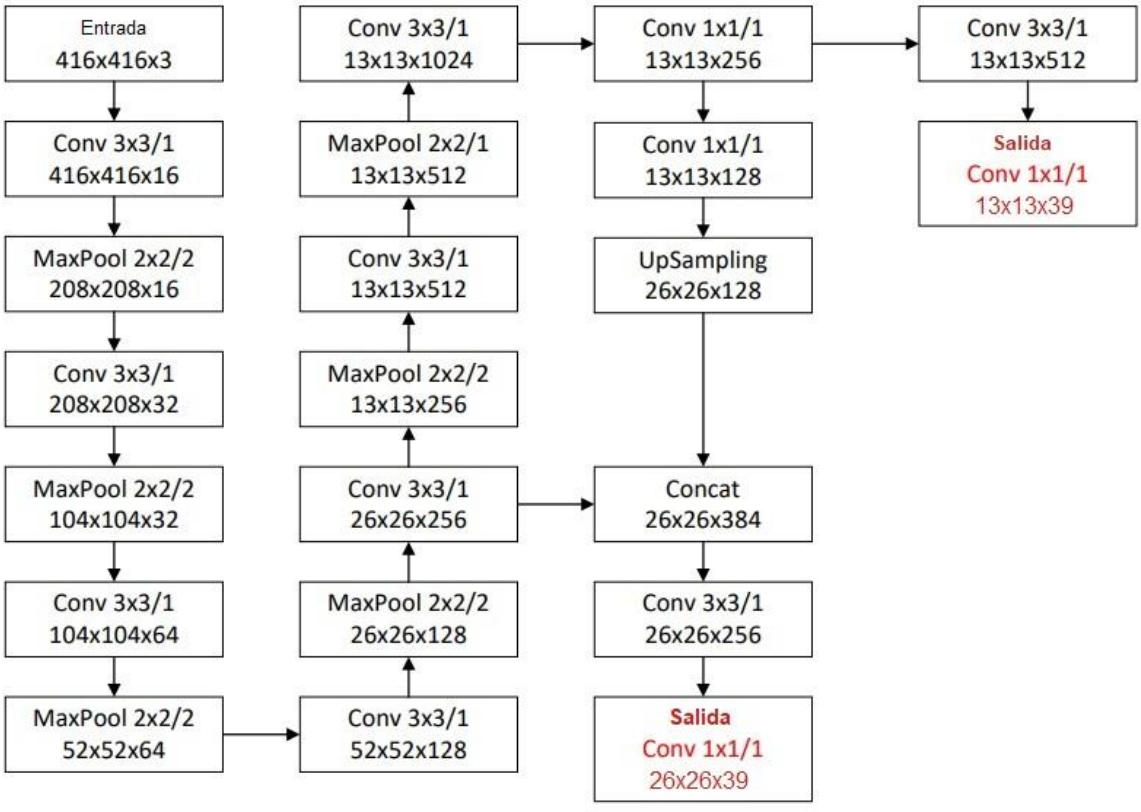


Figura 3.12. Muestra la estructura a bloques de la arquitectura YOLOv3-Tiny con imagen de entrada de tamaño 416x416x3.

3.2.11 Arquitectura YOLOv4

La cuarta generación de YOLO (YOLOv4) se lanzó en abril de 2020. Su contribución se presentó en un artículo presentado por Alexey Bochkovskiy y colaboradores (Bochkovskiy *et al*.,2020).

El modelo YOLOv4 es un modelo optimizado basado en YOLOv3. En comparación con la estructura de la red YOLOv3, la red DarkNet53 en YOLOv4 se cambia a CSPDarkNet53, y la red CSPDarkNet53 se usa como columna vertebral (“backbone”) para la extracción de características. La arquitectura de convolución se basa en la arquitectura DenseNet modificada. Algunas de las ventajas de usar DenseNet es que se resuelven los problemas de desaparición de gradientes decrecientes, el aumento de la propagación hacia atrás, la eliminación del cuello de botella computacional y la mejora del aprendizaje. El cuello (“neck”) se compone de una capa de agrupación piramidal espacial (SPP) y una agregación de rutas PANet. La capa SPP y la agregación de ruta PANet se utilizan para la agregación de características para mejorar el campo receptivo y acortar características importantes de la columna vertebral (“backbone”). Además, la cabeza (“head”) está compuesta por capa de salida YOLO.

El funcionamiento es muy similar a las arquitecturas YOLO anteriores. Primero, la imagen se alimenta a la red CSPDarknet53 para la extracción de características y luego la salida pasa a la red de agregación de rutas PANet para la fusión. Finalmente, la capa YOLO genera los resultados, similar a YOLOv3, YOLOv4 incluye pérdida completa de IOU (CIOU), regularización de bloque de caída y diferentes técnicas de aumento de datos. Además, se utiliza la función de activación “mish” y redes de agregación de rutas modificadas como se observa en la Figura 3.13.

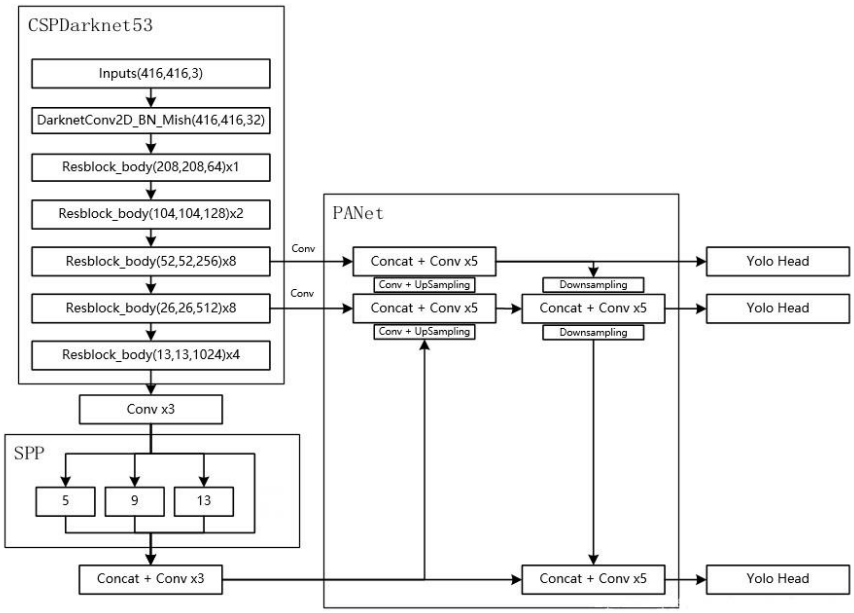


Figura 3.13. Muestra la estructura a bloques de la arquitectura YOLOv4.

3.2.12 Arquitectura YOLOv4-Tiny

La arquitectura de detección YOLOv4-Tiny es una versión simplificada de YOLOv4 que se enfoca en ser más eficiente en términos de velocidad y uso de recursos computacionales, a costa de una menor precisión en la detección de objetos pequeños.

Al igual que YOLOv4, YOLOv4-Tiny se basa en una red neuronal convolucional (CNN) con múltiples capas que se entrenan en datos de imágenes etiquetadas para detectar y clasificar objetos en una imagen dada. YOLOv4-Tiny utiliza la columna vertebral ("backbone") de Darknet53 con capas más ligeras y menos profundas que YOLOv4, lo que hace que el modelo sea más rápido y adecuado para su uso en dispositivos con recursos limitados.

La principal diferencia entre YOLOv4-Tiny y YOLOv4 es que la primera es una versión simplificada y más rápida de la segunda, pero con una precisión de detección de objetos ligeramente menor y dos escalas de detección ("YOLO-Head") como se observa en la Figura 3.14.

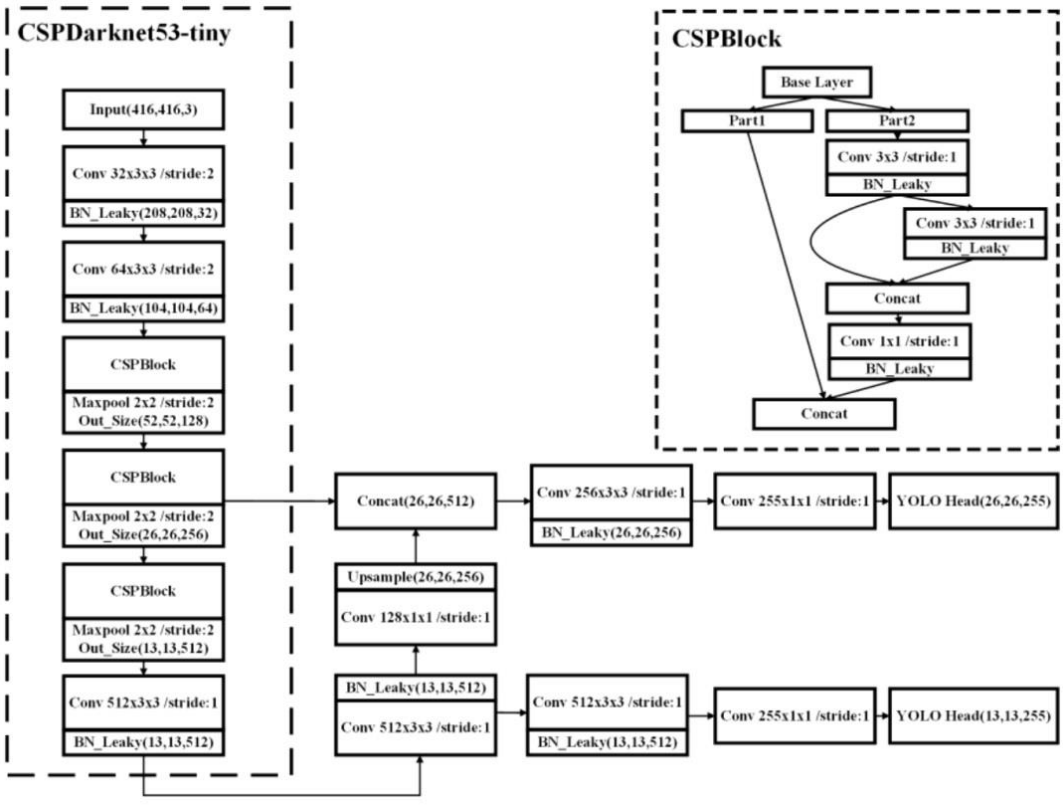


Figura 3.14. Muestra la estructura a bloques de la arquitectura YOLOv4-Tiny.

3.2.13 Arquitectura YOLOv5

YOLOv5 es un modelo de detección de objetos basado en redes neuronales convolucionales (CNN) que utiliza un modelo basado en FPN (“Feature Pyramid Networks”) que mejora la capacidad del modelo para detectar objetos de diferentes tamaños y escalas. Una característica clave de YOLOv5 es su capacidad para ajustar dinámicamente la resolución de entrada de la imagen para mejorar la detección de objetos en diferentes tamaños.

Además, utiliza una técnica de aumentación de datos llamada "mosaic augmentation", que combina varias imágenes en una sola imagen de entrenamiento para mejorar la capacidad del modelo para detectar objetos en entornos complejos. Además, en YOLOv5, se utiliza una variante de FPN llamada "PAN" (“path aggregation network”), que utiliza múltiples vías de características para construir una pirámide de características de alta resolución. PAN permite que el modelo detecte objetos en diferentes escalas y mejora la precisión de la detección de objetos pequeños y grandes. En resumen, el modelo PAN basado en FPN (“Feature Pyramid Networks”) en YOLOv5 mejora la capacidad del modelo para detectar objetos de diferentes tamaños y escalas en la imagen de entrada. Esto aumenta la precisión y eficiencia de la detección de objetos, especialmente en entornos con objetos de diferentes tamaños y escalas.

El diagrama de flujo del proceso de entrenamiento del modelo YOLOv5 se muestra en la Figura 3.15, la cual fue tomada del trabajo de Li *et, al.* (2021). Los autores muestran que el tamaño de la imagen de entrada se redimensiona a m × m píxeles. La imagen se pasa a través de la red YOLOv5 para extraer características, lo que genera mapas de características P0, P1 y P2 con una escala de (m/32) × (m/32) píxeles, (m/16) × (m/16) píxeles y (m/8) × (m/8), respectivamente. Cada píxel de los mapas de características contiene información de tres cuadros delimitadores predichos con valores predichos $\left(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \widehat{P_0}, \widehat{P_c}\right)$. Los mapas de características P0 corresponden al uso de una cuadrícula de 32 × 32 píxeles en la imagen de entrada para generar cuadros de anclaje con tamaños de (116 × 90), (156 × 198) y (373 × 326) en el centro de cada cuadrícula. De la misma manera, los mapas de características P1 corresponden al uso de una cuadrícula de 16 × 16 píxeles para generar cuadros de anclaje con tamaños de (30 × 61), (62 × 45) y (59 × 129). Además, los mapas de características P3 corresponden al uso de una cuadrícula de 8 × 8 píxeles para generar cuadros de anclaje con tamaños de (10 × 13), (16 × 30) y (33 × 23).

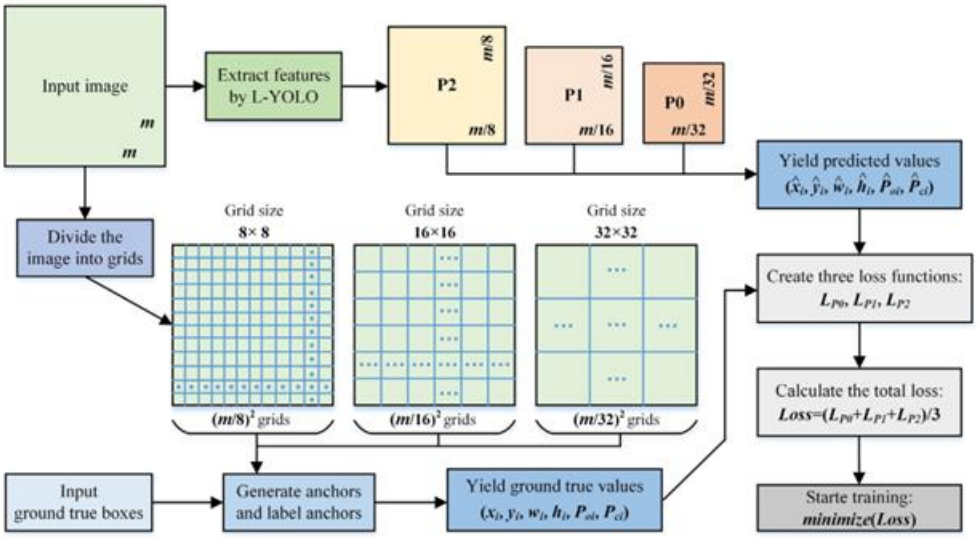


Figura 3.15. Muestra el proceso de entrenamiento del modelo YOLOv5 (Li *et, al.* 2021)

De acuerdo con las coordenadas de ubicación y la categoría de los cuadros verdaderos, cada cuadro de anclaje se etiqueta para obtener los valores reales (x, y, w, h, Po, Pc). La función de pérdida se establece combinando los valores reales con los valores predichos. La función de pérdida total es igual al valor de la suma de las tres funciones de pérdida, construida por la salida de los mapas de características P0, P1 y P2. El proceso de entrenamiento del modelo comienza con la minimización de la función de pérdida. La función de pérdida utilizada en el modelo YOLOv5 se define de la siguiente manera:

$$J_P = 5\sum_{i=0}^{S^2}\sum_{j=0}^{3} 1_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2] + 5\sum_{i=0}^{S^2}\sum_{j=0}^{3} 1_{ij}^{obj}\left[(w_i-\hat{w}_i)^2+\left(h_i-\hat{h}_i\right)^2\right] + \sum_{i=0}^{S^2}\sum_{j=0}^{3} 1_{ij}^{obj}\left[\left(P_{0i}-\hat{P}_{0i}\right)^2\right] + 0.5\sum_{i=0}^{S^2}\sum_{j=0}^{3}\left[\left(1-1_{ij}^{obj}\right)^2\left(P_{0i}-\hat{P}_{0i}\right)^2\right] + \sum_{i=0}^{S^2} 1_{ij}^{obj}\left(P_{ci}-\hat{P}_{ci}\right)^2 \qquad (11)$$

3.2.14 Arquitectura YOLOv5s

La arquitectura YOLOv5 se divide en cuatro versiones: YOLOv5s, YOLOv5m, YOLOv5l y YOLOv5x, siendo YOLOv5s la versión más pequeña y YOLOv5x la más grande y compleja. En comparación con YOLOv5s, las versiones posteriores (m, l y x) tienen más capas y más filtros en cada capa, lo que les permite extraer características más complejas y realizar detecciones de objetos más precisas. Sin embargo, también tienen más parámetros y requieren más tiempo y potencia de cómputo para entrenar y ejecutar. Los elementos principales son la columna vertebral de la red neuronal convolucional ("Backbone"), la cual es una versión modificada de la red "EfficientNet", y está diseñada para extraer características de la imagen con una alta eficiencia computacional. YOLOv5s utiliza una pirámide de características (FPN) que combina diferentes niveles de resolución en la detección de objetos. Esto permite que la red detecte objetos de diferentes tamaños y formas en una imagen. La cabeza de detección de YOLOv5s es la última parte de la red y es responsable de producir las predicciones de objetos. Consiste en múltiples capas convolucionales seguidas de una capa de convolución de 1x1 que produce los resultados de detección, la Figura 3.16 muestra los elementos de esta arquitectura.

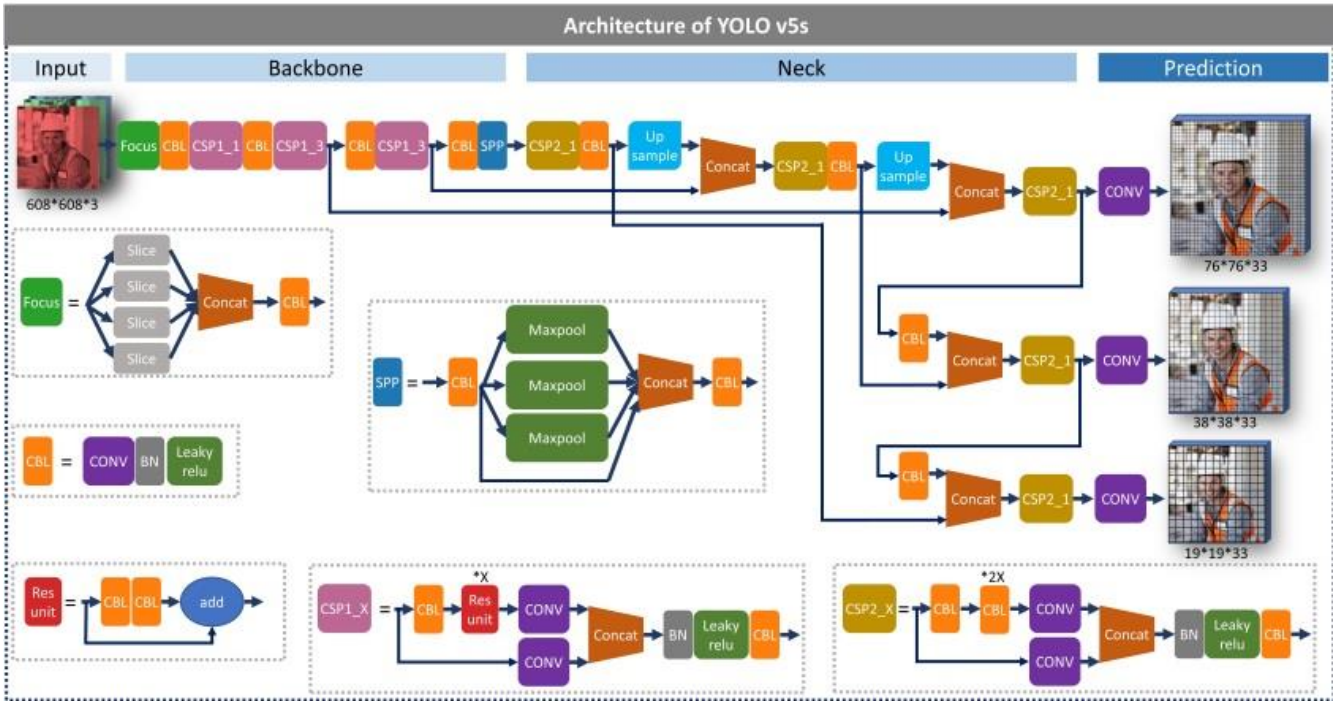


Figura 3.16. Muestra la arquitectura del modelo YOLOv5s (Wang *et, al.* 2021)

4 DESARROLLO

De la revisión del trabajo relacionado presentado en la sección de antecedentes no se encontró información de cuál es la red neuronal que mejor clasifica y detecta la calidad de las manzanas. Por lo tanto, el trabajo de tesis se centró en determinar la arquitectura de red neuronal convolucional (CNN) que permita clasificar y detectar manzanas en sanas y dañadas en el proceso de postcosecha considerando análisis de imágenes en el espectro visible, de las variedades *Red Delicious, Granny Smith, Golden Delicious y Gala*.

4.1 Diseño y recolección de datos

4.1.1 Diseño de un sistema mecatrónico

Para lograr el objeto de estudio fue necesario realizar el diseño y construcción de una banda transportadora equipada con un sistema de visión, que consistió en una estructura de aluminio para soportar una cinta transportadora de PVC grado alimenticio, con un sistema electrónico para el manejo de datos, capaz de transportar manzanas para realizar análisis en masa. El resultado del diseño conceptual y el construido se presentan en la Figura 4.1. La banda transportadora, cuenta con un espacio aislado de condiciones de iluminación externa, este espacio cerrado de acero inoxidable tiene iluminación controlada compuesta por tiras leds que entregan 300 lúmenes por metro. El área de captura es de 200 cm^2 y la altura a la que se encuentran la cámara y la iluminación led es de 30 cm respectivamente. Se utilizó el software libre OpenCV y una cámara logitech modelo C920 con una resolución máxima de 1080p/30fps.

A) Diseño de la banda transportadora B) Captura de imágenes de manzanas

Figura 4.1. Banda transportadora para captura y clasificación de manzanas

La banda transportadora tiene una altura máxima de 110 cm, ancho de 53 cm y 1.4 m de largo. Requiere de una alimentación de 12 V para el motor DC que genera un torque de 3.5 Nm a 100 rpm.

Cuenta con un sistema de transmisión por cadena y tiene una relación de transmisión de 2.7, por lo que reduce la velocidad de giro del rodillo impulsor a 37 rpm, de esta manera la cinta trasportadora corre a 0.3 ms^{-1} y la distancia entre ejes de rodillos es de 112 cm. Además, mediante un mecanismo tensor se puede ajustar a +/- 6 cm para controlar la tensión de la cinta transporta.

4.1.2 Descripción del conjunto de imágenes

La base de datos de imágenes RGB generada para este trabajo es de 4,800 imágenes con dimensión de 800x600 píxeles cada una, que está compuesta por las cuatro variedades de manzanas, organizadas en sanas y dañadas. Los daños considerados para esta base de datos son de origen mecánico, por magulladuras, picaduras, raspaduras y heridas. El 70% de las imágenes se utilizaron para el entrenamiento de las CNN, el 15% para validación y el 15% para prueba. Se reorganizaron las 4,800 imágenes de manzanas en ocho clases: 600 imágenes de manzana Gala sana (Figura 4.2A), 600 imágenes de manzana Gala con daños (Figura 4.2E), 600 imágenes de manzana Golden Delicious sana (Figura 4.2B), 600 imágenes de manzana Golden Delicious con daños (Figura 4.2F), 600 imágenes de manzana Granny Smith sana (Figura 4.2C), 600 imágenes de manzana Granny Smith con daños (Figura 4.2G), 600 imágenes de manzana Red Delicious sana (Figura 4.2D) y 600 imágenes de Red Delicious con daños (Figura 4.2H).

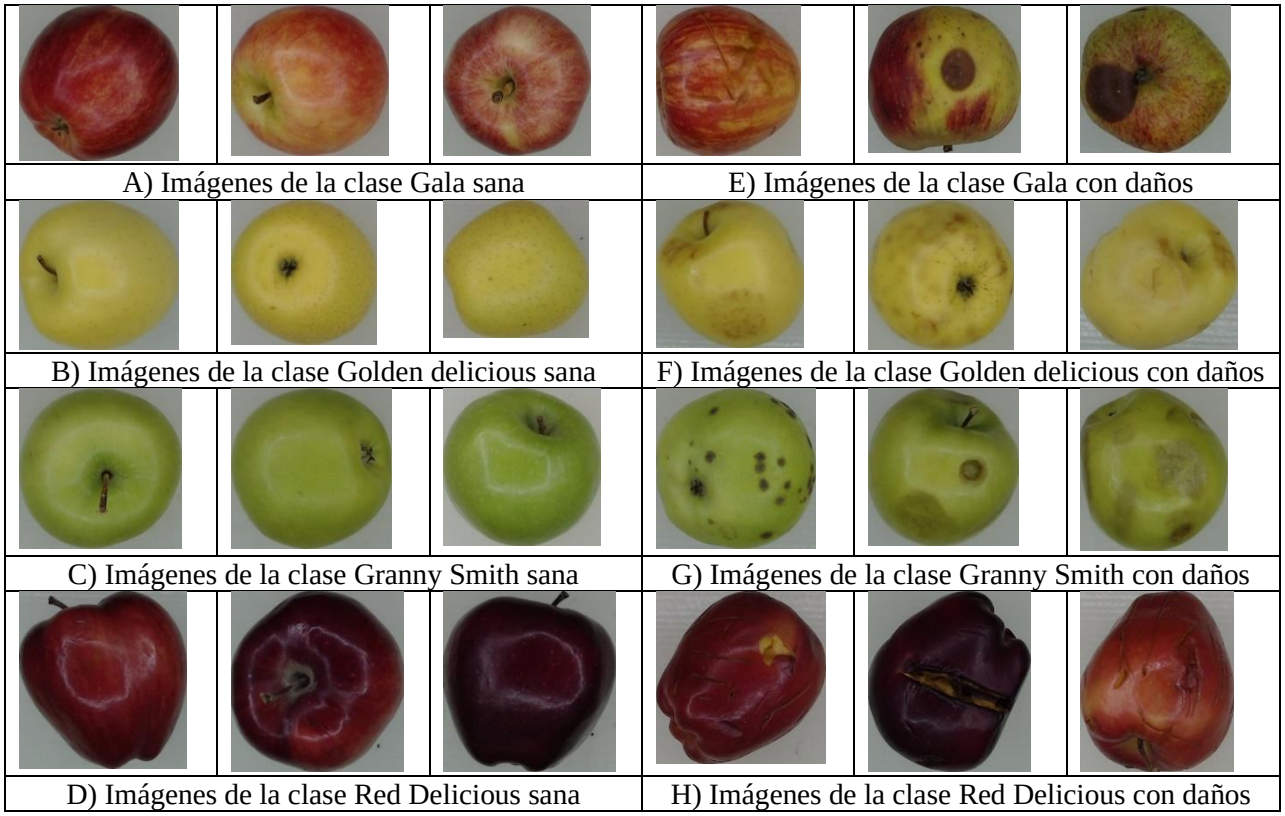


Figura 4.2. Ejemplos de la base de datos de imágenes de manzanas sanas y con daños

4.2 Clasificación

4.2.1 Selección de arquitecturas (CNN) para clasificación

Para encontrar una arquitectura adecuada que permita clasificar manzanas en sanas y dañadas en el proceso de postcosecha, se consideraron las redes neuronales que reportan mejores resultados y se probaron en esta tarea objetivo. A su vez diversos hiperparámetros se probaron con cada arquitectura.

Para lograr la clasificación con métodos no invasivos de las cuatro variedades de manzana en categorías sanas y con daños, como primer paso se propuso desarrollar un sistema de visión en sitio con una arquitectura CNN capaz de realizar esta tarea. Donde, las arquitecturas analizadas fueron: LeNet5 que está basada en la arquitectura propuesta por LeCun *et al*. (1998) y VGG16 propuesta por Simonyan *et al*. (2014). En VGG16 se utilizaron, además, técnicas de aprendizaje por transferencia, específicamente, extracción de características ("feature extraction") y ajuste fino ("fine-tuning"), además de entrenamiento desde cero ("from scratch"). En el aprendizaje por transferencia, se utilizó solamente la arquitectura VGG16 que fue preentrenada con ImageNet (Russakovsky, *et al*., 2015).

La arquitectura de red neuronal convolucional LeNet5 de Yann LeCun (LeCun *et al*., 1998), fue diseñada para el conjunto de datos MNIST del problema de reconocimiento de caracteres escritos a mano, logrando una exactitud en clasificación de 99.2%. Esta arquitectura consta de 2 capas convolucionales, 2 capas de cribado máximo (max pooling) y 3 capas densas (dense). Para la presente se propuso una modificación a las capas densas de la arquitectura LeNet5 de la siguiente manera, a la capa densa1 se le asignaron 256 neuronas con función de activación Relu, a la capa densa2 se le asignaron 84 neuronas con función de activación Relu, y a la capa densa3 se le asignaron 8 neuronas con función de activación SoftMax para poder clasificar las 8 clases (Figura 4.3).

La red neuronal convolucional VGG16 fue propuesta por Simonyan y Zisserman (Simonyan *et al*., 2014). Esta arquitectura participó en el ILSVRC-2014 (ImageNet Large-Scale Visual Recognition Challenge 2014) y alcanzó una precisión del 92.7% quedando entre los 5 primeros en ImageNet (Russakovsky, *et al*., 2015). La arquitectura VGG16 tiene cinco bloques convolucionales (B1,…,B5), cada bloque convolucional es seguido por una capa de agrupación y finalmente se tienen tres capas densas. En este caso se acondicionaron las capas densas de la arquitectura VGG16 de la siguiente manera, a la capa densa1 se le asignaron 4,096 neuronas con función de activación Relu, a la capa densa2 se le asignaron 512 neuronas con función de activación Relu, y a la capa densa3 se le asignaron 8 neuronas con función de activación Softmax para poder clasificar las 8 clases, como se muestra en la Figura 4.3.

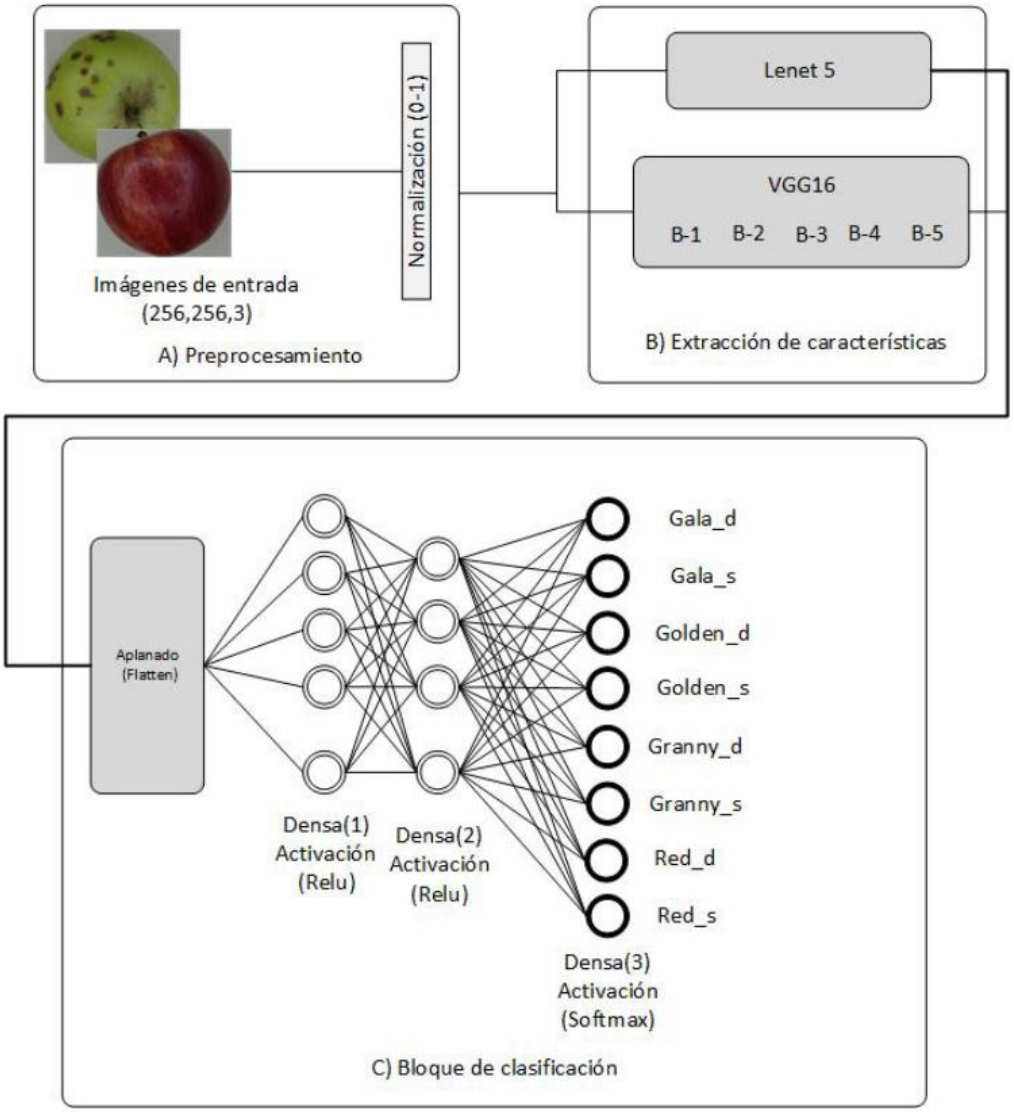


Figura 4.3. Modelos de las arquitecturas LeNet5 y VGG16 propuestas para la extracción de característica y clasificación de manzanas.

4.2.2 Diseño de experimento para clasificación

Se propuso una serie de tratamientos (combinación de red con hiperparámetros) que se utilizaron en la clasificación. Al probarse cada tratamiento se midió su rendimiento. Al finalizar el experimento el tratamiento con mejor rendimiento fue el seleccionado. Los factores involucrados en cada tratamiento fueron: la arquitectura, el número de capas a entrenar, la inicialización de los pesos, el optimizador y la tasa de aprendizaje. Debido a que la cantidad de combinaciones es muy alta (aunque son solo 5 factores, la combinación de los posibles valores en cada factor implica una cantidad de experimentos igual a la combinatoria sin repetición), se decidió realizar el diseño de experimento en dos etapas. En la primera etapa se hace una búsqueda “rápida” entrenando sólo 30 épocas. Esta primera etapa produce las posibles

tasas de aprendizaje. La segunda etapa prueba menos combinaciones, pero con más épocas (500). A continuación, se describe en detalle cada una de las etapas.

4.2.2.1 Búsqueda rápida por rejilla

El objetivo de este experimento es encontrar un conjunto discreto de valores para la tasa de aprendizaje por el método de búsqueda por cuadrícula ("Grid search"). Este método divide un espacio de búsqueda en intervalos discretos uniformes.

Los rangos de ajuste de los hiperparámetros fueron: la tasa de aprendizaje que varió desde 1x10(-08) hasta 0.1 con pasos de 0.001, gamma 0.001 a 0.0001, kernel de 3 y los optimizadores fueron Adam y RMSProp. Se emplearon las arquitecturas LeNet5 y VGG16 con las modificaciones descritas en la Figura 19. Estas fueron entrenadas desde cero y se estableció un número de 30 épocas para cada búsqueda. En esta primera instancia la búsqueda entregó como mejores tasas de aprendizaje las siguientes: 1x10(-03), 1x10(-04), 1x10(-05), 1.5x10(-05) y 1x10(-06).

4.2.2.2 Búsqueda completa por rejilla

El objetivo de este experimento es determinar la arquitectura y sus hiperparámetros que mejor se desempeña en esta tarea. A cada combinación de arquitectura con hiperparámetros se le nombra tratamiento. En el Cuadro 4, se muestran los tratamientos que se probaron. Las arquitecturas son LeNet5 y VGG16. Los optimizadores son Adam y RMSProp. Las tasas de aprendizaje son las cinco que se encontraron en el experimento anterior. Las técnicas de entrenamiento son tres: entrenamiento desde cero, por extracción de características y ajuste fino. El entrenamiento desde cero inicializa de forma aleatoria todos los pesos de la CNN. La extracción de características y el ajuste fino sólo aplica para VGG16. La extracción de características primero inicializa los pesos con una red pre entrenada, segundo, fija los pesos del bloque de extracción de características y tercero habilita para su modificación a las capas densas. El ajuste fino primero inicializa los pesos con una red pre entrenada, y posteriormente fija los pesos de los bloques 1 al 4, habilita el bloque 5 y las capas densas para su modificación. Observe las capas de la red en la Figura 19B y 19C.

Cuadro 4. 40 experimentos para medir rendimiento en clasificación con los tratamientos propuestos.

Identificador	T. E.	Arquitectura	Optimizador	T. A.
1-5	Desde cero	LeNet5	RMSProp	5 posibles*
6-10	Desde cero	LeNet5	Adam	5 posibles*
11-15	Desde cero	VGG16	RMSProp	5 posibles*
16-20	Desde cero	VGG16	Adam	5 posibles*
21-25	Ext. C.	VGG16	RMSProp	5 posibles*
26-30	Ext. C.	VGG16	Adam	5 posibles*
31-35	Ajuste fino	VGG16	RMSProp	5 posibles*
36-40	Ajuste fino	VGG16	Adam	5 posibles*

Ext. C.: Extracción de características. T.A.: Tasa de aprendizaje. T.E.: Técnica de entrenamiento. *Cada renglón condensa 5 unidades experimentales donde se usa una de cinco posibles tasas de aprendizaje: 1x10(-03), 1x10(-04), 1x10(-05), 1.5x10(-05) y 1x10(-06).

En el experimento se realizó un preprocesamiento de imagen que implica un re-escalamiento de cada imagen a 256x256 píxeles, y una normalización (Figura 4.3A). Se establece para el entrenamiento un tamaño de lote por época (Batch size) de 90 imágenes. Los entrenamientos fueron de 500 épocas con una espera ("patience") de 10 épocas para detener el entrenamiento si la perdida no disminuye. La implementación se realizó con software libre, específicamente Python 3.6, OpenCV 4.1.2, tensorflow 2.6.0, Keras y Scikit-learn (Varoquaux *et al*., 2015). e usó la plataforma Google Colaboratory.

4.1.4.3 Métricas de evaluación

Para evaluar el desempeño de cada tratamiento, se utilizan las métricas de: exactitud, recuerdo, puntuación F1, precisión y matriz de confusión; las cuales son definidas en la sección 3.1.7.

4.3 Detección

4.3.1 Propuesta de solución para el problema de detección

La detección de daños en la manzana, en el proceso de embalaje y comercialización, siguen siendo tareas que requieren de mucha mano de obra. Para resolver este problema, un aspecto clave es realizar la detección de forma automática. Sin embargo, detectar con una alta precisión y en tiempo real estos daños, no es una tarea menor, pues es importante distinguir que los cambios de coloración en la textura son una característica estructural clave para determinar si existe algún posible daño y estos cambios en la textura (magulladuras, raspaduras, heridas, etc.) están relacionados directamente con la calidad interna del fruto.

La propuesta de solución para el problema de detección de este trabajo de investigación nivel doctorado, considera que, para lograr una detección favorable y de forma automática de las cuatro variedades de manzana ("Gala", "Golden", "Granny Smith" y "Red Delicious") sanas y dañadas, es necesario analizar, probar y comparar varias arquitecturas de detección de la familia YOLO mediante un diseño de experimento. También, se plantea presentar una serie de modificaciones a la arquitectura YOLO de mejor desempeño, que permita implementar en hardware de bajo costo computacional este algoritmo y así, facilitar la implementación en aplicaciones agroindustriales. Una de las razones principales para considerar las arquitecturas YOLO en este estudio, es que son arquitecturas de una sola etapa que puede detectar múltiples objetos en una sola imagen y proporcionar información de localización precisa para cada objeto. Además, es capaz de manejar imágenes de diferentes tamaños y escalas, lo que hace de utilidad su uso para esta tesis.

4.3.2 Diseño de experimento para detección de manzanas

Un aspecto fundamental para el diseño del experimento es identificar los aspectos principales del problema o tarea a resolver (Kolosoba *et al*., 2020). En este caso, una de las principales dificultades para la detección de daños en manzanas, están relacionadas con las arquitecturas de detección, en las cuales se deben considerar aspectos como: el rendimiento en detección, el tamaño del peso del modelo y la velocidad de detección que permita un desempeño en tiempo real. Para lograr una implementación de detección de daños en manzana en una banda transportadora que cumpla con las condiciones del tiempo real, se necesitan condiciones de GPU, ya que la cantidad del total de parámetros que intervienen en estas tareas, generalmente son del orden de 10^6, por lo cual, se requiere encontrar un algoritmo optimizado.

Una primera pregunta de investigación que se planteó resolver en este trabajo fue la siguiente: ¿Es Yolo, una arquitectura de detección cuyo rendimiento en detección, tamaño del peso del modelo y su velocidad de detección, permiten un desempeño en tiempo real en condiciones de GPU?

Una hipótesis planteada para este diseño de experimento fue que Yolov5s con hiperparámetros obtenidos por búsqueda por cuadrícula y aprendizaje por transferencia, supera en términos de velocidad de detección a otras arquitecturas Yolo entrenadas con hiperparámetros por default (con y sin aprendizaje por transferencia), para la detección de cuatro variedades de manzana (“Gala”, “Golden”, “Granny Smith” y “Red Delicious”) considerando que para esta tarea se pueda distinguir entre manzanas dañadas y sin daños. Así que, para mostrar que la hipótesis es verdadera se planteó aplicar los siguientes tratamientos.

Tratamientos propuestos

- El primer tratamiento es aplicar la arquitectura YOLOV3-Tiny que es una versión reducida de la arquitectura YOLOV3 original, diseñada para ser más liviana y rápida para la detección de objetos en imágenes en tiempo real en dispositivos con recursos limitados. La arquitectura YOLOV3-Tiny se basa en una red neuronal convolucional (CNN) compuesta por varias capas convolucionales y de pooling, seguidas por varias capas completamente conectadas. A diferencia de la arquitectura YOLOV3 completa, que tiene 106 capas, la arquitectura YOLOV3-Tiny solo tiene 23 capas, como se explicó en la sección (3.2.10). Esta arquitectura utiliza la función de activación LeakyReLU para prevenir el problema de saturación de la función de activación ReLU en las capas convolucionales.

- El segundo tratamiento es la arquitectura YOLOv4-Tiny, que utiliza una red neuronal convolucional (CNN) que se compone de 17 capas, incluyendo una capa de entrada y una capa de salida. Otra característica de esta arquitectura es el uso de la función de activación Mish, que se utiliza en lugar de la función ReLU, ya que en la literatura se ha demostrado que es más efectiva en términos de rendimiento y precisión en tareas de detección de objetos (Kolosoba *et al*., 2020). La arquitectura YOLOv4-Tiny utiliza un método de optimización de aprendizaje llamado "SPP-Block", que es una versión reducida del método llamado "Spatial Pyramid Pooling" utilizado en la arquitectura YOLOv4 completa y que se muestra en la sección (3.2.12). Este método reduce la cantidad de cálculos necesarios y, por lo tanto, reduce el tiempo de procesamiento requerido para la detección de objetos.

- El tercer tratamiento es la arquitectura YOLOv5s, la cual se basa en la arquitectura YOLOv4, pero con varias mejoras y optimizaciones para mejorar la velocidad y la precisión en la detección de objetos. En comparación con YOLOv4, YOLOv5s tiene un menor número de capas convolucionales, lo que la hace más liviana y rápida. Además, utiliza la función de activación Swish, que se ha demostrado que es más efectiva que la función de activación ReLU otro aspecto importante es un nuevo enfoque de aumentación de datos llamado "Mosaic Data Augmentation", que utiliza cuatro imágenes diferentes para generar una sola imagen, lo que aumenta la cantidad de datos de entrenamiento y mejora la capacidad de generalización del modelo. También utiliza la técnica de "Grouped Convolutions", que divide los filtros de convolución en grupos para reducir la cantidad de operaciones y mejorar la eficiencia en el procesamiento de imágenes.

Métricas utilizadas

Las métricas que se utilizan para comparar la efectividad de cada tratamiento en el sujeto de estudio serán la precisión, la sensibilidad, la media del promedio de precisión (mAP) y la matriz de confusión, las cuales ya fueron presentadas en la sección (3.1.7 y 3.2.7).

Factorial Completo

El factorial completo que se propuso para dar solución a este experimento, se muestra en el Cuadro 10.

Cuadro 10. Factorial completo para el problema de detección de manzanas

ID	Unidad experimental		Resultados				
	Sujeto	Tratamiento	P	R	mAP@0.5	T.P.	fps
UE-1	Manzanas	Yolov3-Tiny					
UE-2	Manzanas	Yolov4-Tiny					
UE-3	Manzanas	Yolov5s					
UE-4	Manzanas	Yolo-con Propuesta de Modificación					

ID: Identificador. UE: Unidad experimental. P: Precisión. R: Sensibilidad. T.P.: Número total de parámetros de la red. fps: cuadros por segundo.

Búsqueda de hiperparámetros

En la sección 4.1.5 de esta tesis, se mostró que la búsqueda de hiperparámetros no es una tarea menor, principalmente por la cantidad de factores que intervienen. Por esa razón, en este experimento se implementó nuevamente el concepto de búsqueda rápida y búsqueda amplia utilizado en el problema de clasificación de imágenes de manzana.

La búsqueda rápida consistió en probar los tres tratamientos propuestos a pocas épocas (30 épocas), y se determinó que para reducir la complejidad del experimento se considerarían como factores ortogonales los mostrados en el Cuadro 11.

Cuadro11. Factores ortogonales de la experimentación

ID	**Factores Ortogonales**	
	Factor	Valor
1	Imagen	416x416
2	Canales	3
3	Ángulo	0
4	Tinte	0.1
5	Saturación	1.5
6	Exposición	1.5
7	Optimizador	SGD
8	Tamaño de lote	64

ID: Identificador.

En el Cuadro 12, se presenta de forma explícita los factores que fueron utilizados en la búsqueda por rejilla ("grid search") implementada en cada tratamiento.

Cuadro12. Factores utilizados en la búsqueda por rejilla.

ID	Factores de la rejilla			Métricas		
	Tasa de aprendizaje	Momento	Decadencia (Decay)	Precisión	Sensibilidad	mAP @0.5
FR-1	0.000001	0.5	0.00005			
FR-2	0.00001	0.75	0.0005			
FR-3	0.0001	0.9	0.005			
FR-4	0.01	0.95	0.05			

ID: Identificador. FR: Factores de la búsqueda por rejilla.

Se implementó la búsqueda amplia por rejilla, con el objetivo de determinar la arquitectura de detección y sus hiperparámetros que mejor se desempeña en esta tarea. A cada combinación de arquitectura con hiperparámetros se le nombra tratamiento. En el Cuadro 13, se muestran los tratamientos que se probaron para el problema de detección.

Cuadro13. Experimentos realizados para medir el rendimiento en detección con los tratamientos propuestos.

Identificador	Arquitectura	Hiperparámetros	T. E.
M1- M3	Yolov3-Tiny	FR-1 a FR-3	A.T.
M4	Yolov3-Tiny	FR-4	E.S.T.
M5-M7	Yolov4-Tiny	FR-1 a FR-3	A.T.
M8	Yolov4-Tiny	FR-4	E.S.T.
M9-M11	Yolov5-S	FR-1 a FR-3	A.T.
M12	Yolov5-S	FR-4	E.S.T.

T.E.: Técnica de entrenamiento. A.T.: Aprendizaje por transferencia. E.S.T.: Entrenamiento sin aprendizaje por transferencia (Se entrena todo desde cero).

Uno de los objetivos específicos de este trabajo de tesis y que tienen que ver con la detección de daños en manzana, es adecuar la arquitectura de una red neuronal para su uso en sistemas de baja capacidad de cómputo.

Para lograr cumplir con este objetivo, se plantearon una serie de modificaciones a la arquitectura que, con los tratamientos propuestos, presentó el mejor rendimiento en detección. Estas modificaciones se implementaron en un hardware de menor costo computacional para su evaluación. Los experimentos de optimización para la arquitectura ganadora se muestran en el Cuadro 14.

Cuadro 14. Modificaciones de optimización para el tratamiento de mejor desempeño

Identificador	Arquitectura	Modificación
M13	Yolo-Optima1	Se modifica la arquitectura de mejor desempeño para que funcione con sólo dos escalas de detección
M14	Yolo-Optima2	Se modifica la arquitectura de mejor desempeño para que funcione con sólo una escala de detección
M15	Yolo-Optima3	Se modifica la arquitectura de mejor desempeño para que funcione con sólo una escala de detección y con reducción de profundidad

Requerimientos del experimento

Para realizar los experimentos, se utilizó la base de datos de imágenes RGB generada para este trabajo, que es de 4,800 imágenes con dimensión de 800x600 píxeles cada una, y está compuesta por las cuatro variedades de manzanas, organizadas en sanas y dañadas. Esta base fue presentada ampliamente en la sección 4.1.2 de esta tesis. Los entrenamientos para el problema de detección fueron realizados en la plataforma de Google Colab, que cuenta con tarjetas gráficas Tesla P100-PCIE-16GB para entrenamiento y su respectiva validación.

En lo referente a las pruebas de detección, se utilizó el sistema de adquisición de imágenes basado en banda transportadora, el cual fue presentado en la sección 4.1.1. Para implementar la arquitectura YOLO, fue necesario equipar al dispositivo con dos tipos de hardware, los cuales permitieron probar y evaluar el desempeño.

El primer hardware (hardware1) implementado en el dispositivo fue una computadora con procesador Intel® i5 a 2.5GHz y 8GB de memoria RAM. Cuenta con una unidad de procesamiento gráfico (GPU) de la marca “NVIDIA®” modelo “GEFORCE GTX-1650Ti®”, la cual se muestra en la Figura 4.4.

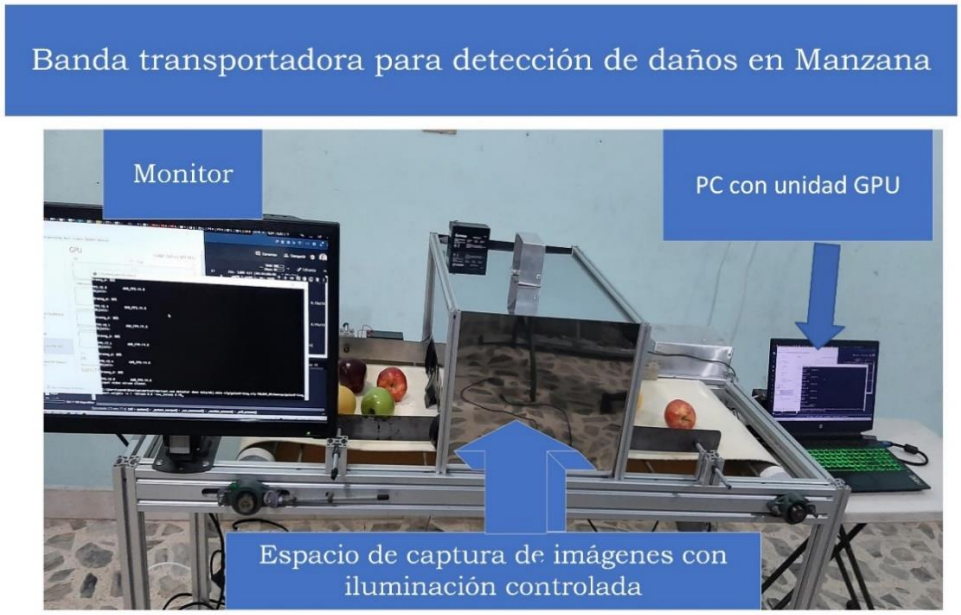


Figura 4.4. Dispositivo equipado con una computadora con procesador Intel® i5 y GPU NVIDIA® GEFORCE GTX-1650Ti®

El segundo hardware (hardware2) implementado en el dispositivo se muestra en la Figura 4.5. Se trata de una tarjeta "Jetson Xavier NX DEVELOPER KIT-SUB®", con las siguientes características:

- CPU: 6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6MB L2 + 4 MB L3.
- GPU: NVIDIA Volta™, 384 NVIDIA®CUDA® cores and 48 Tensor cores.

Figura 4.5. Dispositivo para detección implementado con una tarjeta "Jetson Xavier NX DEVELOPER KIT-SUB®

5 EXPERIMENTOS

5.1 Resultados experimentales del problema de clasificación

Los resultados experimentales de la búsqueda completa por rejilla se presentan en los Cuadros 5 al 8. En el Cuadro 5, se puede observar que la exactitud en el conjunto de prueba se encuentra en el rango de 0.61 a 0.97, y que en los tratamientos (1 al 10) la arquitectura LeNet5 con optimizador RMSProp y tasa de aprendizaje 1x10(-4) es la de mejor desempeño con una exactitud de 0.97. En el Cuadro 6, se puede observar que la exactitud en el conjunto de prueba para los tratamientos (11 al 20) se encuentra en el rango de 0.12 a 0.95, y que la arquitectura VGG16 con optimizador Adam y tasa de aprendizaje 1x10(-5) es la de mejor desempeño con una exactitud de 0.95. El Cuadro 7, que corresponde a los tratamientos (21 al 30) muestra una exactitud en el conjunto de prueba con un rango entre 0.91 a 0.95, y la arquitectura VGG16 con optimizador Adam, tasa de aprendizaje de 1x10(-5) y con extracción de características tiene exactitud de 0.95. Además, el Cuadro 8, que corresponde a los tratamientos (31 al 40) muestra una exactitud en el conjunto de prueba con un rango entre 0.94 a 0.96, y la arquitectura VGG16 con optimizador Adam, tasa de aprendizaje de 1.5x10(-5) y ajuste fino tiene exactitud de 0.96.

Cuadro5. Rendimiento en clasificación de los tratamientos con LeNet5 entrenada desde cero

Hiperparámetros de entrenamiento				Rendimiento en clasificación			
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
1	RMSProp	$1x10^{-03}$	Random	21	0.90	0.66	0.61
2	RMSProp	$1x10^{-04}$	Random	44	0.99	0.98	0.97
3	RMSProp	$1x10^{-05}$	Random	174	0.98	0.94	0.94
4	RMSProp	$1.5x10^{-05}$	Random	131	0.96	0.94	0.92
5	RMSProp	$1x10^{-06}$	Random	226	0.94	0.88	0.87
6	Adam	$1x10^{-03}$	Random	15	0.70	0.58	0.62
7	Adam	$1x10^{-04}$	Random	36	0.99	0.96	0.95
8	Adam	$1x10^{-05}$	Random	65	0.96	0.90	0.89
9	Adam	$1.5x10^{-05}$	Random	64	0.97	0.94	0.93
10	Adam	$1x10^{-06}$	Random	247	0.95	0.91	0.92

Trat: Tratamiento.

Cuadro 6. Rendimiento en clasificación de los tratamientos con VGG16 entrenada desde cero

	Hiperparámetros de entrenamiento				Rendimiento en clasificación		
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
11	RMSProp	$1x10^{-03}$	Random	12	0.10	0.12	0.12
12	RMSProp	$1x10^{-04}$	Random	41	0.98	0.94	0.83
13	RMSProp	$1x10^{-05}$	Random	103	0.98	0.95	0.94
14	RMSProp	$1.5x10^{-05}$	Random	74	0.97	0.96	0.95
15	RMSProp	$1x10^{-06}$	Random	111	0.65	0.57	0.58
16	Adam	$1x10^{-03}$	Random	17	0.11	0.12	0.12
17	Adam	$1x10^{-04}$	Random	46	0.99	0.87	0.84
18	Adam	$1x10^{-05}$	Random	71	0.99	0.97	0.95
19	Adam	$1.5x10^{-05}$	Random	91	0.99	0.94	0.88
20	Adam	$1x10^{-06}$	Random	99	0.77	0.73	0.66

Trat: Tratamiento.

Cuadro 7. Rendimiento en clasificación de los tratamientos con VGG16 y extracción de características

	Hiperparámetros de entrenamiento				Rendimiento en clasificación		
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
21	RMSProp	$1x10^{-03}$	ImageNet	36	0.97	0.94	0.93
22	RMSProp	$1x10^{-04}$	ImageNet	28	0.98	0.93	0.91
23	RMSProp	$1x10^{-05}$	ImageNet	90	0.99	0.95	0.94
24	RMSProp	$1.5x10^{-05}$	ImageNet	51	1	0.95	0.94
25	RMSProp	$1x10^{-06}$	ImageNet	190	1	0.95	0.95
26	Adam	$1x10^{-03}$	ImageNet	22	1	0.95	0.92
27	Adam	$1x10^{-04}$	ImageNet	55	1	0.95	0.95
28	Adam	$1x10^{-05}$	ImageNet	83	1	0.96	0.95
29	Adam	$1.5x10^{-05}$	ImageNet	66	1	0.95	0.95
30	Adam	$1x10^{-06}$	ImageNet	267	1	0.95	0.95

Trat: Tratamiento.

Cuadro 8. Rendimiento en clasificación de los tratamientos con VGG16 y ajuste fino

Hiperparámetros de entrenamiento					Rendimiento en clasificación		
Trat.	Optimizador	Tasa de aprendízaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
31	RMSProp	$1x10^{-03}$	ImageNet	23	0.96	0.78	0.94
32	RMSProp	$1x10^{-04}$	ImageNet	25	0.98	0.94	0.96
33	RMSProp	$1x10^{-05}$	ImageNet	60	1	0.96	0.95
34	RMSProp	$1.5x10^{-05}$	ImageNet	51	1	0.97	0.95
35	RMSProp	$1x10^{-06}$	ImageNet	120	1	0.96	0.95
36	Adam	$1x10^{-03}$	ImageNet	27	1	0.97	0.96
37	Adam	$1x10^{-04}$	ImageNet	36	1	0.98	0.95
38	Adam	$1x10^{-05}$	ImageNet	65	1	0.97	0.96
39	Adam	$1.5x10^{-05}$	ImageNet	70	1	0.97	0.96
40	Adam	$1x10^{-06}$	ImageNet	178	1	0.96	0.95

Trat: Tratamiento.

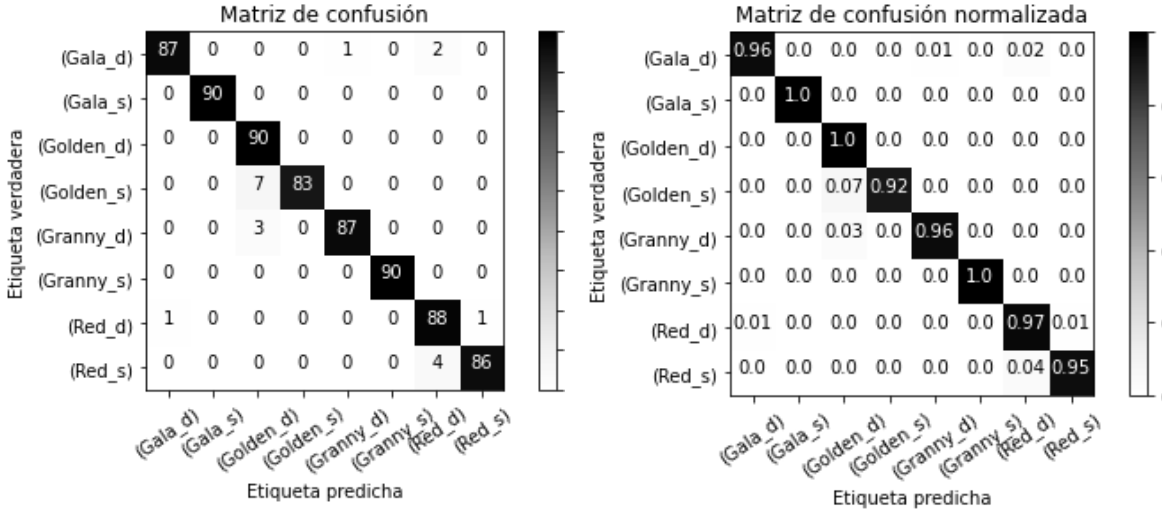
Se determinó que LeNet5 con el tratamiento 2, fue la arquitectura de mejor desempeño pues obtuvo una exactitud de 0.97 (97%). Por lo tanto, es la mejor arquitectura.

En el Cuadro 9, se reportan las métricas de precisión, sensibilidad y F1 de la arquitectura ganadora. En la evaluación de la matriz de confusión, se utilizaron 720 imágenes y el desempeño en clasificación se muestra en la Figura 5.1A sin normalizar y en la Figura 5.1B normalizada.

Cuadro 9. Métricas de desempeño en clasificación de LeNet5 con el tratamiento 2

Categoría	Precisión	Sensibilidad	Puntuación F1	No. de Imágenes
Gala_d	0.99	0.97	0.98	90
Gala_s	1	1	1	90
Golden_d	0.90	1	0.95	90
Golden_s	1	0.92	0.96	90
Granny_d	0.99	0.97	0.98	90
Granny_s	1	1	1	90
Red_d	0.94	0.98	0.96	90
Red_s	0.99	0.96	0.97	90

_d: dañada, _s: sana.



A) Matriz de confusión de LeNet5 con el tratamiento 2

B) Matriz de confusión normalizada de LeNet5 con el tratamiento 2

Figura 5.1. Matriz de confusión de la arquitectura ganadora, donde: _s = sana, _d =dañada.

5.1.2 Discusión de resultados del problema de clasificación

LeNet5 con el tratamiento 2 clasifica bien (100% en la matriz de confusión), las categorías Gala sana, Golden dañada y Granny sana; para las 5 categorías de manzanas restantes, los porcentajes se encuentran entre 92 a 97% (Figura 5.1B). En Fan *et al*., (2020) informan que su arquitectura de CNN entrenada desde cero para clasificar manzanas sanas y dañadas de la variedad Gala, obtuvo un 92% de exactitud con un conjunto de 200 imágenes para la prueba. Además, con técnicas clásicas de visión artificial (Sofu *et al*., 2016) reportan que utilizaron variedades de manzana Granny Smith y Golden Delicious, para clasificar sólo manzanas sanas, y obtuvieron una exactitud del 89%. Kayaalp *et al*., (2020) clasificaron manzanas Gala sanas y dañadas, usaron clasificadores de arquitectura profunda (CNN) y reportaron tasas de exactitud en clasificación de 91.25%.

En contraste, Moallem *et al*., (2017) reportan que para realizar la clasificación utilizaron 16 manzanas Golden sanas y 8 manzanas Golden dañadas y obtuvieron en cada clase un desempeño del 92.5%. Analizando los resultados obtenidos de la matriz de confusión normalizada (Figura 5.1B) se observa también, que la manzana Red Delicious sana se clasifica con un 95% y la Red Delicious dañada con 97%. Además, se muestra que el desempeño en clasificación para la manzana Gala sana es de 100% y para la manzana Gala dañada es de 96%. En Fan *et al*., (2020) se reporta que en su matriz de confusión se alcanzó un 93% para la clase Gala sana y 91% para Gala dañada. En la literatura se informa que Sofu

et al., (2016) obtuvieron un desempeño en clasificación del 93.4% para la categoría Granny Smith sana y 96% para la clase dañada.

Finalmente, se informa que el tamaño de arquitectura de LeNet5 es casi tres veces menor en comparación con VGG16 ya que el número de parámetros (pesos de la red), considerando entrenamientos desde cero, son 47,298,476 y 151,038,280 respectivamente. Esto implica que para un conjunto de imágenes como el presentado en este estudio y que se compone de 4,800 imágenes organizadas en 8 clases, una arquitectura convolucional como LeNet5 entrenada desde cero, es adecuada para cumplir con esta tarea.

5.2 Resultados experimentales del problema de detección

Los resultados experimentales de la búsqueda completa por rejilla para el problema de detección se presentan en el Cuadro 15. Se puede observar que la precisión promedio del rendimiento en detección se encuentra en el rango normalizado de 0.56 a 0.99. La sensibilidad promedio del rendimiento en detección se encuentra en el rango de 0.85 a 0.99. Además, la media del promedio de precisión (mAP@0.5) se encuentra en el rango de 0.7107 a 0.9976. Analizando todos estos resultados, se observa que el tratamiento M11 con la arquitectura YOLOv5s es la de mejor desempeño, pues sus resultados alcanzaron: precisión y sensibilidad promedio de 0.99 respectivamente y un mAP@0.5 de 0.9976.

Cuadro 15. Rendimiento en detección de los tratamientos propuestos

Hiperparámetros de entrenamiento				Rendimiento en detección				
Trat.	Lr	Momentum	Decay	P*	R*	mAP@0.5	T.P.	AVG-fps*
M1	0.000001	0.5	0.00005	0.95	0.97	0.9925	$8.91x10^6$	49.2
M2	0.00001	0.75	0.0005	0.80	0.92	0.9444	$8.91x10^6$	48.3
M3	0.0001	0.9	0.005	0.96	0.98	0.9943	$8.91x10^6$	45.9
M4	0.01	0.95	0.05	0.94	0.96	0.9883	$8.91x10^6$	47.4
M5	0.000001	0.5	0.00005	0.98	0.98	0.9912	$10.2x10^6$	46.4
M6	0.00001	0.75	0.0005	0.92	0.97	0.9831	$10.2x10^6$	45.8
M7	0.0001	0.9	0.005	0.98	0.99	0.9936	$10.2x10^6$	45.2
M8	0.01	0.95	0.05	0.96	0.97	0.9892	$10.2x10^6$	43.6
M9	0.000001	0.5	0.00005	0.87	0.85	0.9217	$7.04x10^6$	59.4
M10	0.00001	0.75	0.0005	0.56	0.95	0.7107	$7.04x10^6$	60.4
M11	0.0001	0.9	0.005	0.99	0.99	0.9976	$7.04x10^6$	63.2
M12	0.01	0.95	0.05	0.88	0.90	0.9704	$7.04x10^6$	62.5

Trat: Tratamiento. Lr: Tasa de aprendizaje. P: Promedio de precisión. R: Promedio de sensibilidad. T.P.: Número total de parámetros de la red. AVG-fps*: Promedio de cuadros por segundo medidos en la banda transportadora utilizando el hardware1(procesador Intel® i5 y GPU NVIDIA® GEFORCE GTX-1650Ti®).

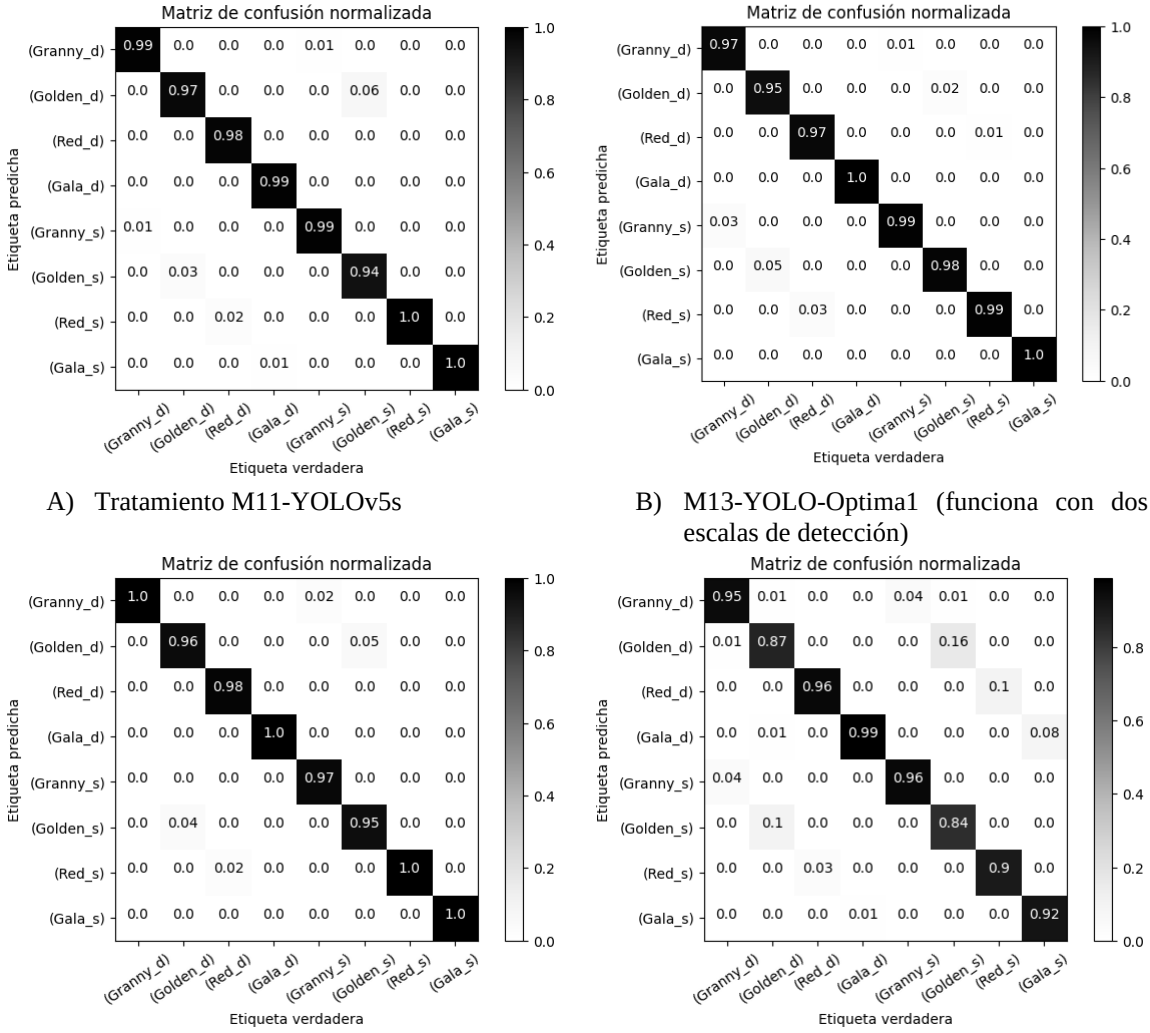
Las modificaciones de optimización para el tratamiento de mejor desempeño (M11), que permitan adecuar la arquitectura M11 para su uso en sistemas de baja capacidad de cómputo y que fueron planteadas en el Cuadro 14, se implementaron en un hardware de menor costo computacional para su evaluación (hardware2). Los experimentos de optimización y su rendimiento en detección se muestran en el Cuadro 16.

Cuadro 16. Rendimiento en detección de las modificaciones de optimización

Identificador	Arquitectura	P	R	mAP@0.5	T.P.	AVG-fps*
M13	Yolo-Optima1	0.98	0.97	0.9917	5.24 x10^6	30.3
M14	Yolo-Optima2	0.98	0.98	0.9908	4.78 x10^6	35.5
M15	Yolo-Optima3	0.94	0.94	0.9776	1.20 x10^6	37.0

P: Promedio de precisión. R: Promedio de sensibilidad. T.P.: Número total de parámetros de la red. AVG-fps*: Promedio de cuadros por segundo medidos en la banda transportadora utilizando el hardware2 ("Jetson Xavier NX DEVELOPER KIT-SUB®).

Las matrices de confusión del rendimiento en detección de la arquitectura ganadora y las modificaciones de optimización se muestran en la Figura 5.2.



A) Tratamiento M11-YOLOv5s

B) M13-YOLO-Optima1 (funciona con dos escalas de detección)

C) M14-YOLO-Optima2 (funciona con sólo una escala de detección)

D) M15-YOLO-Optima3 (funciona con una escala de detección y reducción de profundidad)

Figura 5.2. Muestra el rendimiento en detección de la arquitectura ganadora (A) y las modificaciones de optimización (B), (C) y (D).

En la Figura 5.3, se presenta una muestra de la detección para algunas variedades de manzana con la arquitectura Yolov5s y los hiperparámetros que corresponden al tratamiento M11.

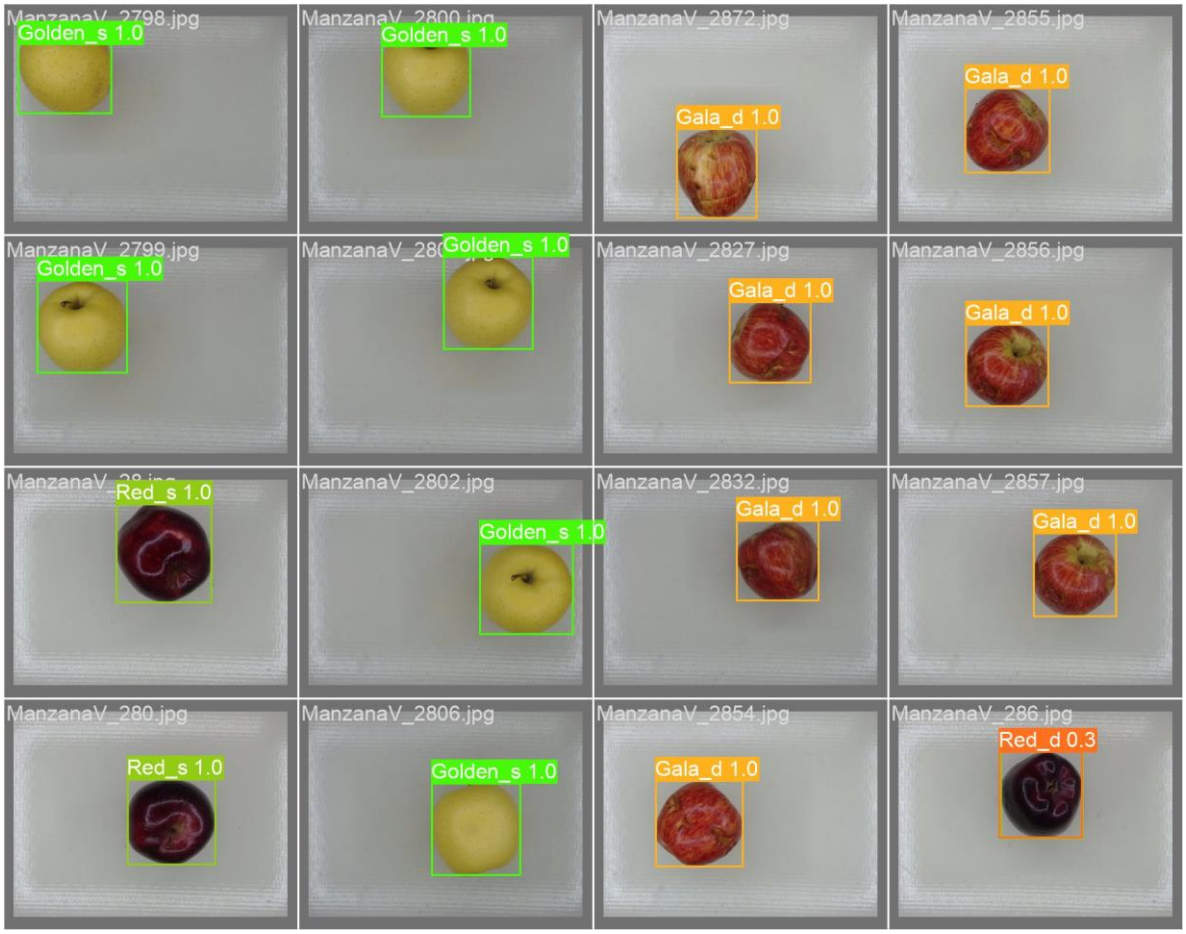


Figura 5.3. Muestra el desempeño en detección de la arquitectura ganadora (M11).

En el contexto de la detección de objetos, el tamaño de la red es un factor importante para considerar, ya que influye en el tiempo de procesamiento necesario para detectar objetos en una imagen o un video. Para calcular el tamaño de la red, se consideró el número total de parámetros (valores reportados en los Cuadros 15 y 16). En el anexo B de esta tesis se dan detalles a manera de ejemplo de cómo se calculó este valor.

Para mostrar los valores las curvas precisión-sensibilidad y puntuación F1 de la arquitectura ganadora (M11) y las modificaciones de optimización (M13), (M14) y (M15). En el anexo C de esta tesis se presentan las gráficas correspondientes.

5.2.2 Discusión de resultados del problema de detección

YoloV5s con los hiperparámetros del tratamiento M11 (Cuadro 15) detecta bien (100% en la matriz de confusión), las categorías Gala sana y Red sana; para las 6 categorías de manzanas restantes, los porcentajes se encuentran entre 94 a 99% (Figura 5.2A), y el mAP@0.5 fue de 0.9976.

Los resultados del tratamiento M11 (Cuadro 15), superan a los reportados en Valdez *et al*., (2020) en donde informan que su arquitectura Yolov3 para un conjunto de entrenamiento de 452 imágenes de manzanas, organizada en 226 imágenes de manzanas sanas y 226 imágenes de manzanas con defectos; obtuvo un mAP@0.5 de 0.7430 para la arquitectura YoloV3 y un mAP@0.5 de 0.6959 para la arquitectura SSD ("single shot detector").

En el trabajo de Chen *et al*., (2021) reportan que, de acuerdo con sus resultados, la arquitectura YoloV4 entrega un mAP@0.5 para detección de manzana del 97,13%, una tasa de sensibilidad del 90% y una velocidad de detección de 51 cuadros por segundo. El trabajo realizado en esta tesis muestra que el tratamiento M11 (Cuadro 15), además de superar el rendimiento en detección reportado por Chen *et al*., (2021), alcanza una velocidad de detección de 63.2 cuadros por segundo.

La detección de manzana con una arquitectura YoloV5s se reporta en el trabajo de Yan *et al*., (2021) en donde sus resultados indican un rendimiento en detección con los siguientes valores: sensibilidad, precisión, mAP@0.5 y puntuación F1, que fueron de 91.48%, 83.83%, 86.75% y 87.49%, respectivamente. En contraste, los resultados del tratamiento M11 (Cuadro 15), reportados en este trabajo, también superan a los de Yan *et al*., (2021).

Los sistemas de detección de manzanas en tiempo real requieren arquitecturas optimizadas para hacer que los robots recolectores detecten manzanas de forma rápida y precisa en un entorno complejo (Wang *et al*., 2022). Para medir el tamaño de una red Yolo, fue necesario contar el número total de parámetros de la red. El número total de parámetros depende de la arquitectura de la red, así como, del tamaño de la imagen de entrada, el número de capas de convolución, el tamaño de los filtros de convolución, entre otros. El tratamiento M11 tiene un total de parámetros del orden de 7.04×10^6 y para optimizar esta arquitectura se realizaron las modificaciones (M13), (M14) y (M15) reportadas en el Cuadro 16, con un total de parámetros de 5.24×10^6, 4.78×10^6 y 1.20×10^6 respectivamente.

Finalmente, Wang *et al*., (2022) informan que su arquitectura optimizada Des-YOLOv4 tiene resultados del mAP@0.5 para detección de manzanas de 97,13%, una tasa de sensibilidad de 90% y una velocidad de detección de 51 cuadros por segundo evaluados en un hardware Intel (R) Core (TM) i7-9700KF CPU @ 3.60 GHz, NVIDIA RTX2080 GPU (8 GB VRAM) y 32 GB RAM. La arquitectura más ligera de este trabajo corresponde al tratamiento M15, esta obtuvo un mAP@0.5 de 97.76% y una velocidad de detección de 37 cuadros por segundo, evaluados en un hardware de menor capacidad (tarjeta Jetson Xaxier Nx, descrita en la Figura 4.5).

6 CONCLUSIONES

Se cumplió el objetivo de este trabajo de tesis. Se desarrolló un sistema mecatrónico para la clasificación visual y detección de daños en manzana (*Malus domestica*).

En la etapa de clasificación, se ha demostrado que la red neuronal LeNet5, entrenada desde cero con el optimizador RMSProp y una tasa de aprendizaje $1x10^{-4}$, logra clasificar de manera precisa las manzanas sanas y dañadas de las variedades Red Delicious, Granny Smith, Golden Delicious y Gala, con una exactitud del 97%. Al analizar la matriz de confusión, se puede inferir que esta arquitectura funciona de manera efectiva al 100% para clasificar las categorías Gala sana, Golden dañada y Granny Smith sana. Para las cinco categorías restantes, el porcentaje de precisión varía entre 92 y 97%. Este hallazgo es significativo, ya que la norma oficial mexicana (NMX-FF-061-SCFI-2003) permite que hasta 10 % de las manzanas de cualquier lote no cumplan con los requisitos establecidos. Por lo tanto, con el fin de implementar un sistema que cumpla con la norma oficial mexicana, la arquitectura encontrada resulta adecuada.

En la etapa de detección de manzanas, se llevaron a cabo experimentos utilizando diferentes arquitecturas e hiperparámetros. Tras analizar los resultados, se concluye que la arquitectura YoloV5s, con los hiperparámetros del tratamiento M11, demostró el mejor desempeño. Esta arquitectura logró una detección precisa de las categorías Gala sana y Red sana, con un 100% de acierto según la matriz de confusión. Para las seis categorías restantes de manzanas, los porcentajes de detección oscilaron entre el 94% y el 99% (Figura 4B). Además, el valor de mAP@0.5 obtenido fue de 99.76%, lo cual indica un alto grado de precisión en la detección. Estos resultados demuestran la efectividad de la arquitectura YoloV5s en la detección de manzanas, destacando su capacidad para identificar y clasificar de manera precisa las diferentes categorías.

Un segundo hallazgo en esta tesis fue que las propuestas de modificación a la arquitectura de mejor desempeño (Yolov5s), permitieron encontrar arquitecturas optimizadas con una reducción en tamaño del 25.57% (arquitectura Yolo-Optima1), 32.39% de reducción (arquitectura Yolo-Optima2), y 82.96% de reducción (arquitectura Yolo-Optima3), manteniendo en estas modificaciones valores altos de precisión y sensibilidad, cuyos promedios están ente 94 y 99% respectivamente.

Se comprobó que las arquitecturas optimizadas para detección de objetos se pueden implementar en tarjetas de hardware de bajo costo computacional, pues se alcanzó una velocidad de detección de 37 cuadros por segundo en el hardware2 y de 63.2 cuadros por segundo en el hardware1.

7 TRABAJO A FUTURO

Para mejorar la precisión de los cuadros delimitadores en la detección de manzanas sanas y dañadas, además del filtro de Kalman, existen otras técnicas que se proponen probar como trabajo a futuro, y se mencionan a continuación.

Filtros de seguimiento, los filtros de seguimiento, como el filtro de Kalman extendido (EKF), el filtro de partículas (PF) y el filtro de correlación de características (DCF), se utilizan para rastrear y estimar la trayectoria de un objeto a medida que se mueve en un video. Estos filtros aprovechan la información de cuadros anteriores y ayudan a mejorar la precisión del cuadro delimitador al proporcionar una estimación más coherente del objeto a lo largo del tiempo.

Los enfoques de detección en múltiples etapas: Algunos algoritmos de detección de objetos utilizan enfoques de múltiples etapas, donde la detección inicial se realiza en una etapa inicial y luego se refinan los resultados en etapas posteriores. Estas etapas pueden incluir técnicas como el refinamiento de cuadros delimitadores mediante regresión, donde se ajustan los cuadros delimitadores iniciales para mejorar su precisión.

Los métodos de posprocesamiento, como el ajuste de no máximos (NMS), se utilizan para eliminar las detecciones duplicadas y fusionar cuadros delimitadores superpuestos. Esto ayuda a obtener una salida más limpia y precisa al eliminar las redundancias y seleccionar los cuadros delimitadores más relevantes.

Las redes neuronales recurrentes (RNN), en particular las variantes como las redes neuronales LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit), se pueden utilizar para modelar la información temporal en secuencias de imágenes. Al capturar el contexto a lo largo del tiempo, estas redes pueden mejorar la precisión de los cuadros delimitadores al tener en cuenta la información de fotogramas anteriores y futuros.

Estas son solo algunas de las técnicas utilizadas para mejorar la precisión de los cuadros delimitadores en algoritmos de detección de objetos. Respecto a las técnicas que se pueden utilizar para la detección de objetos, actualmente se encuentran los Transformers, los cuales son una buena posibilidad como alternativa para detección de daños en frutos de manzana.

Los Transformers son una arquitectura de redes neuronales que ha demostrado un gran éxito en tareas de procesamiento de lenguaje natural, como la traducción automática y el procesamiento del lenguaje natural. Sin embargo, también se ha aplicado con éxito en problemas de visión por computadora, incluida la detección de objetos.

Una de las implementaciones más conocidas de los Transformers en la detección de objetos es la red neuronal llamada DETR (Detection Transformer). DETR es un enfoque que utiliza una arquitectura completamente basada en Transformers para realizar la detección y la asociación de objetos en una imagen. A diferencia de los métodos de detección de objetos tradicionales, como YOLO o Faster R-CNN, DETR no utiliza regiones de interés (RoIs) ni propuestas de regiones, sino que trata la detección de objetos como un problema de clasificación y regresión directa. Sin embargo, es importante tener en cuenta que los Transformers pueden ser computacionalmente más costosos que las arquitecturas más tradicionales, por lo que pueden requerir más recursos y tiempo de entrenamiento.

Sin embargo, además de los Transformers, existen varias técnicas comunes utilizadas en la detección de objetos y con las que se debe explorar la aplicación estas en la detección de enfermedades de manzana. Algunas de estas técnicas incluyen las siguientes.

Métodos basados en regiones de interés (RoI): Estos métodos, como R-CNN, Fast R-CNN y Faster R-CNN, utilizan regiones propuestas generadas previamente para identificar las áreas de interés en una imagen y luego clasificar y ajustar los cuadros delimitadores en esas regiones. Estos enfoques se basan en la extracción de características utilizando CNN y la posterior clasificación y regresión en las regiones propuestas.

Métodos de fusión de características: Estos métodos, como Feature Pyramid Networks (FPN) y Path Aggregation Network (PANet), fusionan características de diferentes escalas y niveles de abstracción para mejorar la detección de objetos en imágenes con objetos de diferentes tamaños. Estas técnicas ayudan a capturar información contextual y detalles finos en diferentes escalas.

Aprendizaje basado en atención: Los métodos basados en atención, como Non-local Neural Networks (NLNet) y CBAM (Convolutional Block Attention Module), se centran en capturar relaciones de largo alcance entre las características de una imagen. Estas técnicas utilizan mecanismos de atención para asignar importancia a diferentes partes de la imagen, lo que puede mejorar la capacidad de la red para detectar objetos en contextos complejos.

Métodos de posprocesamiento: Los métodos de posprocesamiento, como el ajuste de no máximos (NMS) y el filtrado por confianza, se utilizan para eliminar las detecciones redundantes y mejorar la precisión. Estas técnicas ayudan a seleccionar los cuadros delimitadores más relevantes y eliminar las detecciones superpuestas o de baja confianza.

Estas son solo algunas de las técnicas utilizadas en la detección de objetos que marcan el rumbo a seguir. La elección de la técnica depende del contexto y los requisitos específicos del problema, en este caso este trabajo de tesis se analizó el fruto de la manzana en la etapa de postcosecha, pero también se podría analizar directamente en los cultivos.

8 REFERENCIAS

Al-Marakeby, A., Aly, A. A., and Salem, F. A. (2013). Fast quality inspection of food products using computer vision. *International Journal of Advanced Research in Computer and Communication Engineering [1]*, 2.

Baneh, N. M., Navid, H., and Kafashan, J. (2018). Mechatronic components in apple sorting machines with computer vision. *Journal of Food Measurement and Characterization*, 12(2), 1135-1155.CONAGUA, Comisión Nacional del Agua (2016)

Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, 96-107.

Bargoti, S., and Underwood, J. (2017). Deep fruit detection in orchards. In 2017 IEEE international conference on robotics and automation (ICRA) (pp. 3626-3633). IEEE.

Bhargava, A., and Bansal, A. (2021). Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University-Computer and Information Sciences*, 33(3), 243-257.

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Chen, M. C., Cheng, Y. T., and Liu, C. Y. (2022). Implementation of a Fruit Quality Classification Application Using an Artificial In-telligence Algorithm. Sensors and Materials, 34(1), 151-162.

Chen, W., Zhang, J., Guo, B., Wei, Q., & Zhu, Z. (2021). An apple detection method based on des-YOLO v4 algorithm for harvesting robots in complex environment. Mathematical Problems in Engineering, 2021, 1-12.

Crisosto, C. H. (1999). Optimum procedures for ripening stone fruit. Management of Fruit Ripening. University of California, Davis, Postharvest Horticulture Series, 9, 28-30.

Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., ... and Huang, W. (2020). Online detection of defective apples using computer vision system combined with deep learning methods. *Journal of Food Engineering*, 286, 110102.

FAO, Organización de las Naciones Unidas para la Alimentación y la Agricultura (2010). Norma para las manzanas (CODEX STAN 299-2010). Codex alimentarius. Washington.https://www.fao.org/fao-who-codexalimentarius/codex-texts/list-standards/es/ (octubre 2021).

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27-48.

Hameed, K., Chai, D., and Rassau, A. (2018). A comprehensive review of fruit and vegetable classification techniques. Image and Vision Computing, 80, 24-44.

Hou, S., Feng, Y., & Wang, Z. (2017). Vegfru: A domain-specific dataset for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 541-549).

Hu, G., Zhang, E., Zhou, J., Zhao, J., Gao, Z., Sugirbay, A., ... and Chen, J. (2021). Infield Apple Detection and Grading Based on Multi-Feature Fusion. Horticulturae, 7(9), 276.

Huang, Z., Zhang, P., Liu, R., and Li, D. (2021). Immature apple detection method based on improved Yolov3. ASP Transactions on Internet of Things, 1(1), 9-13.

Hussain, I., He, Q., & Chen, Z. (2018). Automatic fruit recognition based on dcnn for commercial source trace system. Int. J. Comput. Sci. Appl, 8(2/3), 01-14.

Jiang, B., He, J., Yang, S., Fu, H., Li, T., Song, H., and He, D. (2019). Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. *Artificial Intelligence in Agriculture*, 1, 1-8.

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147, 70-90.

Katarzyna, R., & Paweł, M. (2019). A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. Applied Sciences, 9(19), 3971.

Kavdır, I., and Guyer, D. E. (2008). Evaluation of different pattern recognition techniques for apple sorting. *Biosystems engineering*, 99(2), 211-219.

Kayaalp, K., and Metlek, S. (2020). Classification of robust and rotten apples by deep learning algorithm. Sakarya University Journal of Computer and Information Sciences, 3(2), 112-120.

Kolosova, T., and Berestizhevsky, S. (2020). Supervised machine learning: optimization framework and applications with SAS and R. CRC Press.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

Lavrač, N., Flach, P., and Zupan, B. (1999). Rule evaluation measures: A unifying view. In International Conference on Inductive Logic Programming (pp. 174-185). Springer, Berlin.

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

Li, G., Huang, X., Ai, J., Yi, Z., & Xie, W. (2021). Lemon-YOLO: An efficient object detection method for lemons in the natural environment. IET Image Processing, 15(9), 1998-2009.

Li, Q., Wang, M., and Gu, W. (2002). Computer vision-based system for apple surface defect detection. *Computers and electronics in agriculture*, 36(2-3), 215-223.

Liu, Z., He, Y., Cen, H., and Lu, R. (2018). Deep feature representation with stacked sparse auto-encoder and convolutional neural network for hyperspectral imaging-based detection of cucumber defects. *Transactions of the ASABE*, 61(2), 425-436.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Lu, R., Pothula, A., Mizushima, A., Van Dyke, M., and Zhang, Z. (2018). Device and system for sorting apples in the orchard. U.S. Patent no. 9,919,345. Washington, DC.

Lu, S., Lu, Z., Aok, S., & Graham, L. (2018). Fruit classification based on six layer convolutional neural network. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP) (pp. 1-5). IEEE.

Lu, Y. (2016). Food image recognition by using convolutional neural networks (cnns). *arXiv preprint arXiv:1612.00983*.

Lu, Y., & Lu, R. (2017). Non-destructive defect detection of apples by spectroscopic and imaging technologies: a review. *Transactions of the ASABE*, 60(5), 1765-1790.

Moallem, P., Serajoddin, A., and Pourghassem, H. (2017). Computer vision-based apple grading for golden delicious apples based on surface features. *Information processing in agriculture*, 4(1), 33-40.

Monta, M., Kondo, N., Ting, K. C., Giacomelli, G. A., Mears, D. R., Kim, Y., and Ling, P. P. (1998). Harvesting end-effector for inverted single truss tomato production systems. *Journal of the Japanese Society of Agricultural Machinery*, 60(6), 97-104.

Mureşan, H., & Oltean, M. (2017). Fruit recognition from images using deep learning. arXiv preprint arXiv:1712.00580.

Nguyen, C. N., Van-Linh, L., Le, P. H., Ho, H. T., and Nguyen, C. N. (2022). Early detection of slight bruises in apples by cost-efficient near-infrared imaging. International Journal of Electrical and Computer Engineering, 12(1), 349.

NMX-FF-061-SCFI-2003 (2003). Productos agrícolas no industrializados para consumo humano - fruta fresca- manzana (Malus pumila Mill) - (Malus domestica Borkh) - especificaciones.
https://www.dof.gob.mx/nota_detalle.php?codigo=704678&fecha=17/02/2003 (mayo 2021).

Patel, K. (7 de septiembre de 2019), Redes neuronales convolucionales: una guía para principiantes.https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022

Patino-Saucedo, A., Rostro-Gonzalez, H., & Conradt, J. (2018). Tropical fruits classification using an alexnet-type convolutional neural network and image augmentation. In International Conference on Neural Information Processing (pp. 371-379). Springer, Cham.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825-2830.

Raschka S. and Mirjalili V. (2019). Aprendizaje automático con python. Segunda edición. Marcombo. España. 616p.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Redmon, J., and Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

Sachin, C., Manasa, N., Sharma, V., and AA, N. K. (2019, November). Vegetable classification using you only look once algorithm. In 2019 International Conference on Cutting-edge Technologies in Engineering (ICon-CuTE) (pp. 101-107). IEEE.

Sakib, S., Ashrafi, Z., Siddique, M., & Bakr, A. (2019). Implementation of fruits recognition classifier using convolutional neural network algorithm for observation of accuracies for various hidden layers. arXiv preprint arXiv:1904.00783.

Sharma, M. R. (2015). Recognition of Anthracnose Injuries on Apple Surfaces using YOLOV 3-Dense. International Journal of New Practices in Management and Engineering, 4(02), 08-14.

SIAP, Sistema de Información Agropecuaria y Pesquera (2020). Panorama agroalimentario 2020. Secretaría de Agricultura y Desarrollo Rural. Ciudad de México.

https://nube.siap.gob.mx/gobmx_publicaciones_siap/pag/2020/Atlas-Agroalimentario-2020 (Octubre 2021).

Siddiqi, R. (2019). Automated apple defect detection using state-of-the-art object detection techniques. SN Applied Sciences, 1(11), 1345.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sofu, M. M., Er, O., Kayacan, M. C., and Cetişli, B. (2016). Design of an automatic apple sorting system using machine vision. *Computers and Electronics in Agriculture*, 127, 395-405.

Steinbrener, J., Posch, K., & Leitner, R. (2019). Hyperspectral fruit and vegetable classification using convolutional neural networks. Computers and Electronics in Agriculture, 162, 364-372. doi:10.1016/j.compag.2019.04.019

Sun, K., Li, Y., Peng, J., Tu, K., and Pan, L. (2017). Surface Gloss Evaluation of Apples Based on Computer Vision and Support Vector Machine Method. *Food Analytical Methods*, 10(8), 2800-2806.

Sun, Y., Lu, R., Lu, Y., Tu, K., and Pan, L. (2019). Detection of early decay in peaches by structured-illumination reflectance imaging. *Postharvest Biology and Technology*, 151, 68-78.

Throop JA, Aneshansley DJ, Upchurch BL, and Anger B (2001). Apple orientation on two conveyors: performance and predictability based on fruit shape characteristics. Trans ASAE 44(1):99–109.

Valdez, P. (2020). Apple defect detection using deep learning based object detection for better post harvest handling. arXiv preprint arXiv:2005.06089.

Wang, S. H., & Chen, Y. (2020). Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. Multimedia Tools and Applications, 79(21), 15117-15133.

Wang, Z., Jin, L., Wang, S., and Xu, H. (2022). Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. Postharvest Biology and Technology, 185, 111808.

Wang, Z., Wu, Y., Yang, L., Thirunavukarasu, A., Evison, C., and Zhao, Y. (2021). Fast personal protective equipment detection for real construction sites using deep learning approaches. Sensors, 21(10), 3478.

Willis, R.H., Lee, T.H. (1990). Fisiología y manipulación de frutas y hortalizas post-recolección, Ed. Acribia, Zaragoza

Wu A., J. Zhu and T. Ren (2020) Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. Computers & Electrical Engineering 81:106454, https://doi.org/10.1016/j.compeleceng.2019.106454

Xu, R., Li, C., Paterson, A. H., Jiang, Y., Sun, S., and Robertson, J. S. (2018). Aerial images and convolutional neural network for cotton bloom detection. *Frontiers in plant science*, 8, 2235.

Yan, B., Fan, P., Lei, X., Liu, Z., and Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sensing, 13(9), 1619.

Zeng, G. (2017). Fruit and vegetables classification system using image saliency and convolutional neural network. In 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC) (pp. 613-617). IEEE.

Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., and Wang, S. H. (2019). Image based fruit category classification by 13-layer deep convolutional neural

network and data augmentation. Multimedia Tools and Applications, 78(3), 3613-3632.

Zhang, Z., Pothula, A. K., and Lu, R. (2017). Economic evaluation of apple harvest and in-field sorting technology. Transactions of the ASABE, 60(5), 1537.

Zhang, B., Huang, W., Li, J., Zhao, C., Fan, S., Wu, J., and Liu, C. (2014). Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, 62, 326-343.

Zhang, B., Huang, W., Gong, L., Li, J., Zhao, C., Liu, C., and Huang, D. (2015). Computer vision detection of defective apples using automatic lightness correction and weighted RVM classifier. *Journal of Food Engineering*, 146, 143-151.

Zhang, W., Zhao, D., Gong, W., Li, Z., Lu, Q., and Yang, S. (2015). Food image recognition with convolutional neural networks. In 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom) (pp. 690-693). IEEE.

Zhu, L., Li, Z., Li, C., Wu, J., and Yue, J. (2018). High performance vegetable classification from images based on alexnet deep learning model. *International Journal of Agricultural and Biological Engineering*, *11*(4), 217-223. DOI: 10.25165/j.ijabe.20181104.2690

A. Anexo. Norma oficial mexicana de calidad de la manzana NMX-FF-061-SCFI-2003

La norma **NMX-FF-061-SCFI-2003** denominada como "*PRODUCTOS AGRÍCOLAS NO INDUSTRIALIZADOS PARA CONSUMO HUMANO - FRUTA FRESCA - MANZANA (Malus pumila Mill) - (Malus domestica)*". Establece los requisitos de calidad y tolerancias que deben cumplir las manzanas para comercialización y consumo humano.

En esta norma mexicana se encuentran las definiciones que se consideraron en esta investigación para poder diferenciar entre una manzana sana y una con daño (defecto).

En el caso de la madurez de la manzana para consumo humano, la norma la define como: "*la etapa de la fruta, en la cual se ha desarrollado color, sabor, textura y aroma característicos de la variedad; y que cumple con los requisitos mínimos de sólidos solubles totales (azúcares) y de resistencia a la penetración*".

La fruta sana es "*fruta libre de enfermedades, heridas, pudriciones, daños producidos por insectos u otras plagas, libre de insectos vivos o muertos o sus larvas*". Además, también se indica que la fruta puede presentar defectos que afectan su apariencia o utilidad, y que el origen de estos daños (defectos) pueden ser de origen genético, fisiológico o mecánico, los cuales ocurren como resultado de anormalidades hereditarias o como efecto de condiciones ambientales desfavorables durante el crecimiento y maduración de la fruta, también la mala manipulación del fruto en la etapa de cosecha y postcosecha contribuyen a que se generen los daños.

Los daños que se pueden presentar son el entomológico, que son causados por las actividades propias de los insectos que incluyen alimentación, oviposición y picaduras. El daño microbiológico es causado por hongos, bacterias, levaduras o virus. Los daños mecánicos que son causados por la manipulación inadecuada de las frutas durante la cosecha y postcosecha de estas, siendo este último uno de los más comunes y que ocasiona las denominadas magulladuras y picaduras, las magulladuras provocan el reblandecimiento o manchas en la cáscara o pulpa y son ocasionadas por golpes o compresiones. Las picaduras, son heridas más o menos profundas ocasionadas por medios mecánicos o por depredadores como aves, roedores u otros.

Dentro de la clasificación de daños se pueden presentar los de origen meteorológico que son ocasionados por diversos fenómenos atmosféricos, como granizo, lluvia, viento y heladas. El daño por dióxido de carbono (CO_2) es provocado por un inadecuado sistema de refrigeración para la fruta que es almacenada bajo condiciones de atmósfera controlada, transportada en vehículos con mala ventilación o cuando es tratada con CO_2 en postcosecha.

El daño por sol, es un daño producido por temperaturas altas mezclado con radiación ultravioleta. Algunos otros daños que se pueden observar en las manzanas son: raspaduras, heridas cicatrizadas, heridas no cicatrizadas, grietas, quemaduras de sol, escalado superficial, por mencionar algunos.

Para mostrar los efectos de los daños en la manzana, en la figura 2 se muestran algunos ejemplos. Además, en el Cuadro 2 se presentan algunas fisiopatías que afectan la calidad de la manzana.

	
Magulladura	Congelamiento
	
Daño por sol	Daño por CO_2
	
Herida	Escalado superficial
	
Daño por raspadura	Herida por uña en la manipulación

Figura 2. Muestra algunos daños presentes en las manzanas.

Cuadro2. Muestra las principales fisiopatías presentes en la manzana

Fisiopatías	**Variedad de manzana que más afecta**	**Síntomas externos**
Congelamiento	Todas	Apariencia húmeda, ablandamiento, coloración oscura, mayor sensibilidad a podredumbre, daños mecánicos y deshidratación.
Corazón pardo	Granny Smith	Nunca llega a presentar síntomas externos.
Deshidratación	Golden Delicious, Gala y Red Delicious	Pérdida de brillo, rugosidad y arrugamiento
Lenticelosis	Gala y Fuji	Lenticelas oscurecidas, negras o pequeñas manchas marrones superficiales.
Escaldadura Superficial	Red Delicious y Granny Smith	Manchas morrones, oscurecimiento del color. La superficie también se puede tornar rugosa o con picaduras
Decaimiento de Senescencia	Red Delicious	La piel se torna marrón y puede rajarse.
Mancha amarga	Red Delicious, Granny Smith, Braeburn y Fuji	Depresiones circulares y oscuras en la piel. Manchas de color oscuro o verde-amarronado.
Plara	Red Delicious, Granny Smith y Fuji	Manchas irregulares de margen definido, color oscuro o negro, de textura seca y ligeramente deprimidas
Harinosidad	Red Delicious y Gala	Los daños se presentan en la pulpa
Partidura	Gala	Separación de las células y ocurrencia de partidura.
Corazón acuoso	Red Delicious	Pardeamiento con posterior descomposición y fermentación del tejido
Toxicidad por difenilamina	Todas	Puntos marrones con aspecto chorreado en la piel, similares a la escaldadura superficial pero más oscuras y brillantes

B. Anexo. Cálculo del número total de operaciones para la arquitectura YOLOv3-Tiny.

El número total de parámetros para esta arquitectura se calcula sumando los parámetros de cada capa. A continuación, se muestra las operaciones realizadas en cada capa de dicha arquitectura con una entrada (416,416,3):

Capa 1: Convolutional -> 16 filtros, tamaño de kernel 3x3 -> 448 parámetros
Capa 2: Maxpool -> tamaño de kernel 2x2 -> 0 parámetros
Capa 3: Convolutional -> 32 filtros, tamaño de kernel 3x3 -> 4640 parámetros
Capa 4: Maxpool -> tamaño de kernel 2x2 -> 0 parámetros
Capa 5: Convolutional -> 64 filtros, tamaño de kernel 3x3 -> 18496 parámetros
Capa 6: Maxpool -> tamaño de kernel 2x2 -> 0 parámetros
Capa 7: Convolutional -> 128 filtros, tamaño de kernel 3x3 -> 73856 parámetros
Capa 8: Maxpool -> tamaño de kernel 2x2 -> 0 parámetros
Capa 9: Convolutional -> 256 filtros, tamaño de kernel 3x3 -> 295168 parámetros
Capa 10: Maxpool -> tamaño de kernel 2x2 -> 0 parámetros
Capa 11: Convolutional -> 512 filtros, tamaño de kernel 3x3 -> 1180160 parámetros
Capa 12: Maxpool -> tamaño de kernel 2x1 -> 0 parámetros
Capa 13: Convolutional -> 1024 filtros, tamaño de kernel 3x3 -> 4719616 parámetros
Capa 14: Convolutional -> 256 filtros, tamaño de kernel 1x1 -> 262400 parámetros
Capa 15: Convolutional -> 512 filtros, tamaño de kernel 3x3 -> 1180160 parámetros
Capa 16: Convolutional -> 39 filtros, tamaño de kernel 1x1 -> 20055 parámetros
Capa 17: YOLO -> 2535 parámetros
Capa 18: Route -> 0 parámetros
Capa 19: Convolutional -> 128 filtros, tamaño de kernel 1x1 -> 16512 parámetros
Capa 20: Upsample -> 0 parámetros
Capa 21: Route -> 0 parámetros
Capa 22: Convolutional -> 256 filtros, tamaño de kernel 3x3 -> 1180160 parámetros
Capa 23: Convolutional -> 39 filtros, tamaño de kernel 1x1 -> 20055 parámetros
Capa 24: YOLO -> 2535 parámetros

De esta manera, el número total de parámetros de la arquitectura YOLOV3-Tiny se calcula sumando todos los parámetros de todas las capas, como se muestra a continuación:

448 + 4640 + 18496 + 73856 + 295168 + 1180160 + 4719616 + 262400 + 1180160 + 20055 + 2535 + 16512 + 1180160 + 20055 + 2535 = 8918903

Finalmente, se concluye que esta arquitectura tiene en total 8,918,903 parámetros.

C. Anexo. Gráficas de precisión-sensibilidad y puntuación F1.

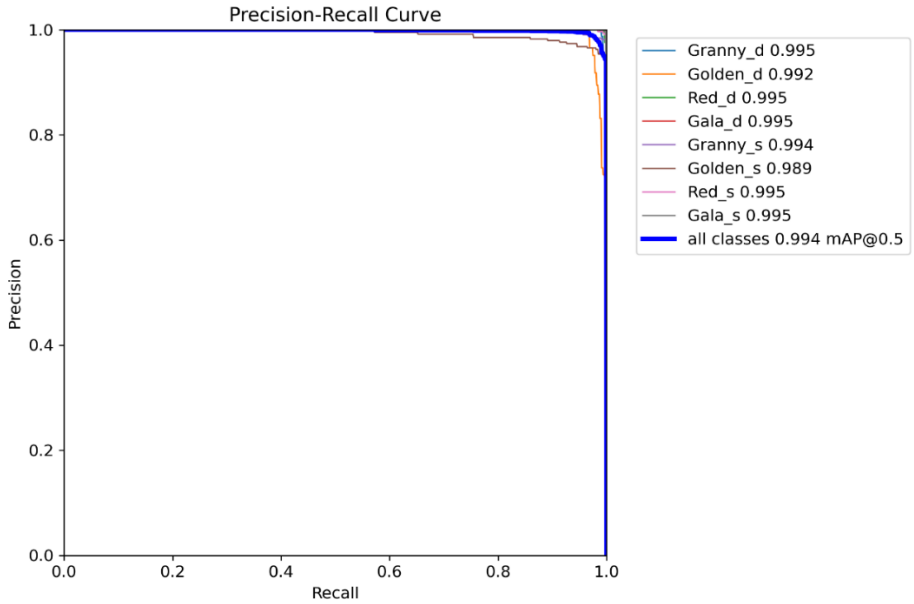


Figura C1. Gráfica de precisión-sensibilidad del tratamiento M11

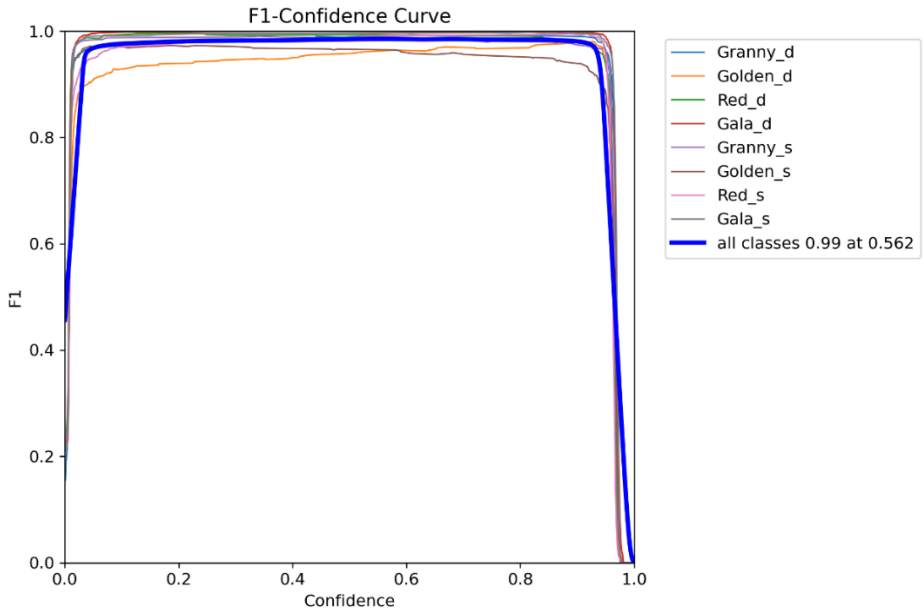


Figura C2. Gráfica de la puntuación F1 del tratamiento M11

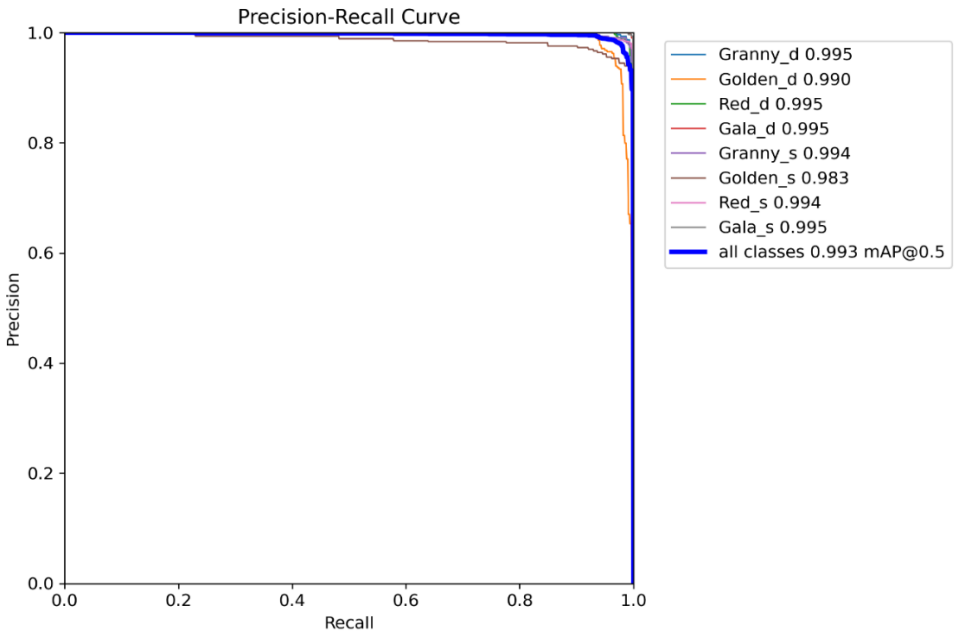


Figura C3. Gráfica de precisión-sensibilidad de la modificación de optimización M13

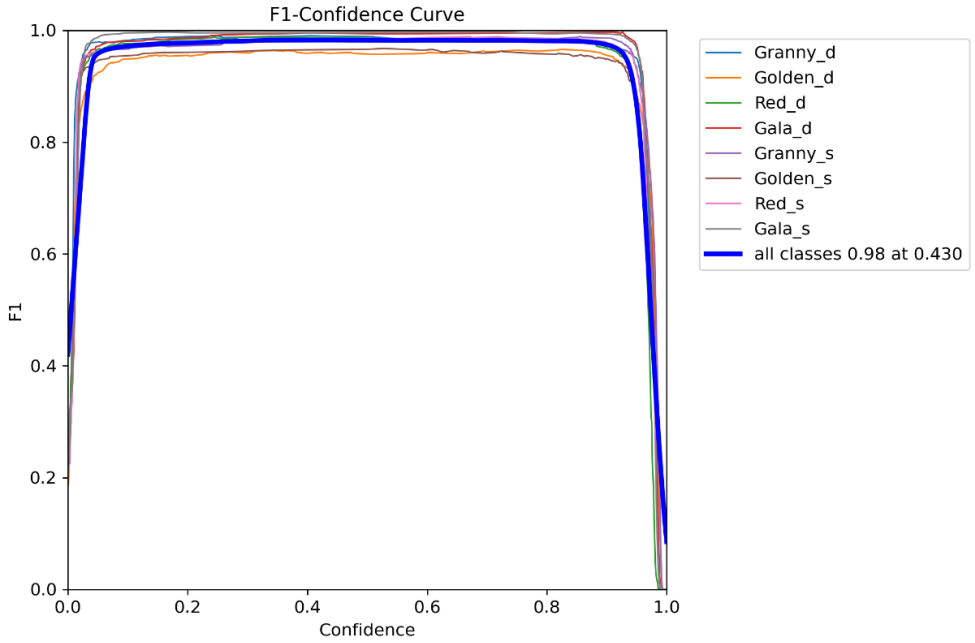


Figura C4. Gráfica de la puntuación F1 de la modificación de optimización M13

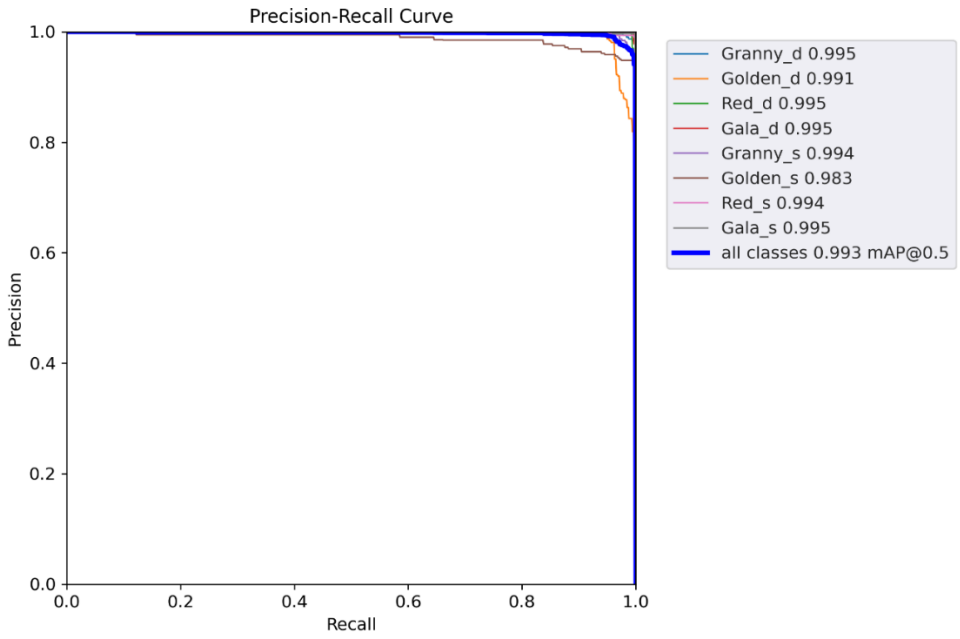


Figura C5. Gráfica de precisión-sensibilidad de la modificación de optimización M14

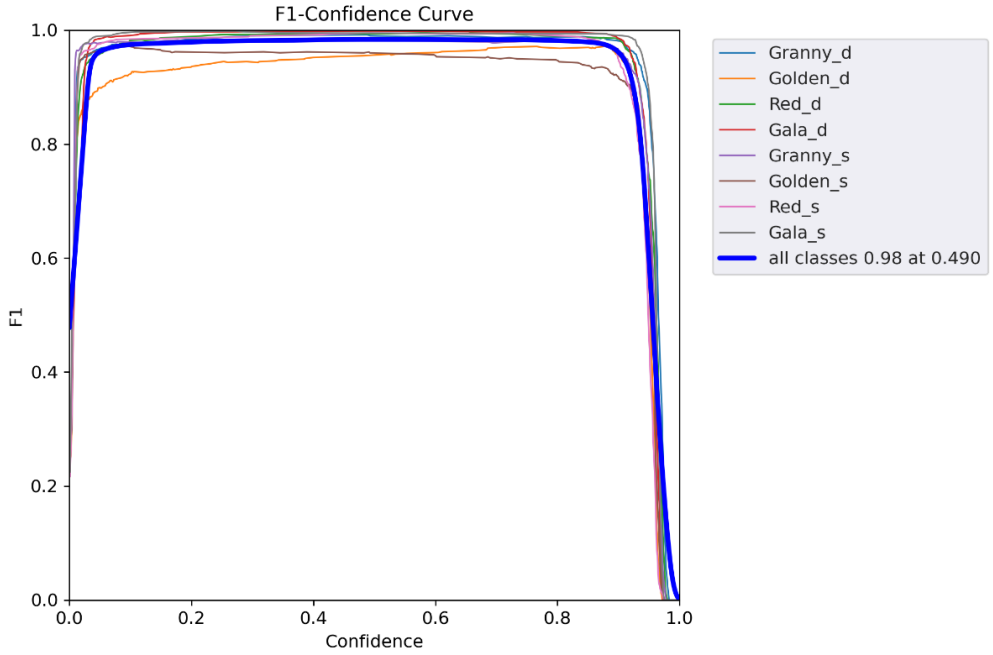


Figura C6. Gráfica de la puntuación F1 de la modificación de optimización M14

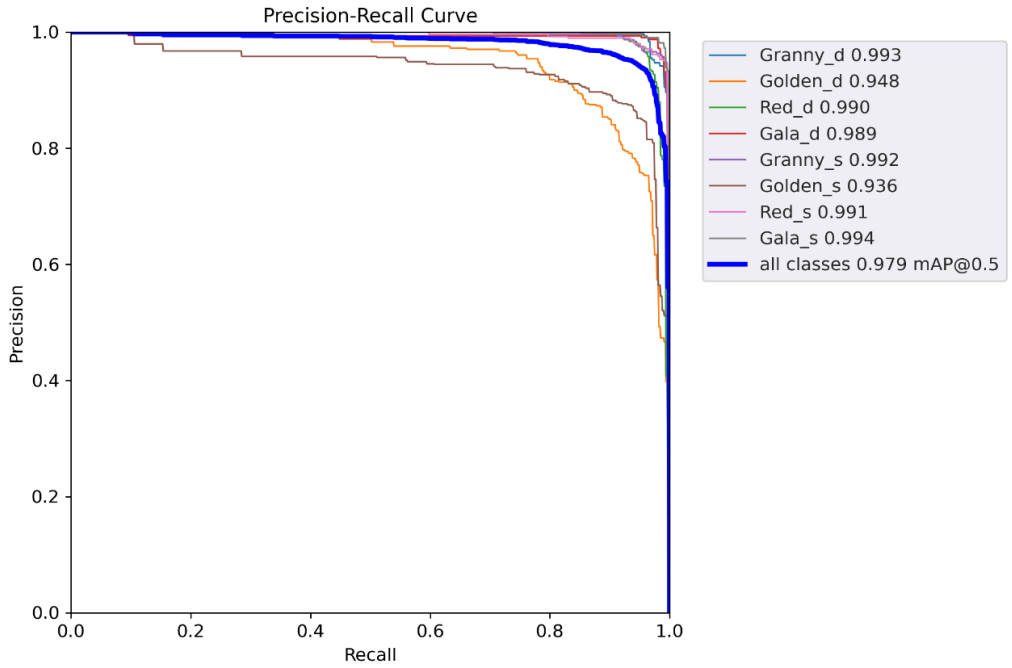


Figura C7. Gráfica de precisión-sensibilidad de la modificación de optimización M15

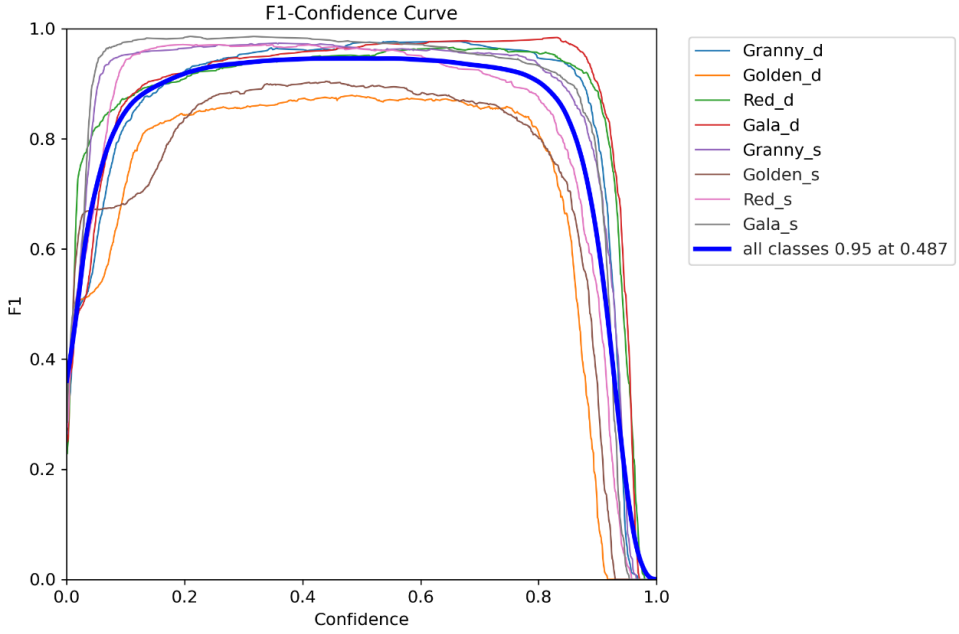


Figura C8. Gráfica de la puntuación F1 de la modificación de optimización M15

D. Anexo. Artículo de revista publicado.

CLASIFICACIÓN DE MANZANAS CON REDES NEURONALES CONVOLUCIONALES

CLASSIFICATION OF APPLES WITH CONVOLUTIONAL NEURONAL NETWORKS

Juan C. Olguín-Rojas[1,2]*, Juan I. Vasquez-Gomez[1], Gilberto de J. López-Canteñs[2] y Juan C. Herrera-Lozada[1]

[1]Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Ciudad de México, México. [2]Universidad Autónoma Chapingo, Departamento de Ingeniería Mecánica Agrícola, Chapingo, Texcoco, Estado de México, México.

**Autor de correspondencia* (jolguinr@chapingo.mx)

RESUMEN

Actualmente, en puntos de venta y en empresas agroindustriales de México, la clasificación de manzanas (*Malus domestica*) la realizan personas de forma manual, lo que genera deficiencias en la calidad del producto. Estos problemas se pueden reducir con la implementación de equipos de visión en sitio equipados con algoritmos de aprendizaje automático. En este estudio se analizaron varias arquitecturas de red neuronal convolucional (CNN) y se seleccionó una que permite clasificar manzanas en sanas y dañadas en el proceso en postcosecha. Las variedades utilizadas fueron Red Delicious, Granny Smith, Golden Delicious y Gala. Se comparó la exactitud de las CNN LeNet5 y VGG16. Se realizó una serie de tratamientos (combinación de red con hiperparámetros) que se utilizaron para la clasificación del objeto de estudio. Al probarse cada tratamiento se midió su rendimiento. Al finalizar, el tratamiento con mejor rendimiento fue LeNet5 entrenada desde cero con el optimizador RMSProp, que obtuvo una exactitud del 97 %.

Palabras clave: *Malus domestica*, clasificación, LeNet5, VGG16.

SUMMARY

Nowadays, in points of sale and in agro-industrial companies in Mexico, the classification of apples (*Malus domestica*) is carried out manually by people, which generates deficiencies in the quality of the product. These problems can be reduced with the implementation of in site vision equipment with machine learning algorithms. In this study, several convolutional neuronal network (CNN) architectures were analyzed and one of those was selected that allows apples to be classified into healthy and damaged in the postharvest process. The varieties used were Red Delicious, Granny Smith, Golden Delicious and Gala. The accuracy of the LeNet5 and VGG16 CNNs was compared. A series of treatments (combination of network with hyperparameters) was performed that were used for the classification of the object of study. As each treatment was tested, its performance was measured. At the end, the treatment with the best performance was LeNet5 trained from scratch with the RMSProp optimizer, which obtained an accuracy of 97 %.

Index words: *Malus domestica*, classification, LeNet5, VGG16.

INTRODUCCIÓN

En México, la manzana se produce en más de 10 entidades federativas, siendo el estado de Chihuahua el principal productor de la zona norte con aproximadamente 624,696 toneladas al año (SIAP, 2020). Durante los procesos de cosecha y postcosecha de manzana, la correcta clasificación es fundamental, ya que los frutos son catalogados por su grado de maduración y calidad, y los precios de mercado están determinados por dichas inspecciones (Fan *et al*., 2020).

La mala clasificación de la manzana en postcosecha evita cumplir con los estándares descritos en la norma oficial mexicana (SE, 2003). Es importante distinguir que, en las líneas de clasificación en el proceso de postcosecha, se considera suficiente con detectar manzanas que cumplen con el estándar de calidad, lo cual permite seleccionar las frutas en 'aprobadas' (sanas) y 'no aprobadas (FAO, 2010).

Actualmente, la clasificación en postcosecha es limitada ya que las técnicas de medición requeridas son inaccesibles para la mayoría de los agricultores, debido a que las mediciones convencionales están basadas en pruebas fisicoquímicas para determinar la consistencia, daños, el grado de madurez o de inocuidad de un fruto y usualmente se realizan en laboratorio. Una posible solución a este problema es la aplicación de técnicas de visión artificial que no son invasivas y no requieren de laboratorios especializados.

La visión artificial ha mostrado eficacia en la clasificación de diversas frutas, como se muestra en los siguientes estudios: Zhang *et al*. (2017) desarrollaron una cosechadora de manzanas autopropulsada que cuenta con una máquina clasificadora en campo y, a través de visión computacional, tomaron decisiones de acuerdo con el color, tamaño, forma y defectos, su sistema mostró bajo costo y rápida velocidad de inspección; en el trabajo de Lu *et al*. (2017) se mejoró la detección de los defectos

Recibido: 31 de diciembre de 2021
Aceptado: 23 de mayo de 2022

DOI: https://doi.org/10.35196/rfm.2022.3.369

de la manzana al incorporar la superficie del fruto y la presencia de tallos y cálices; Fan *et al*. (2020) usaron una arquitectura de aprendizaje profundo basada en redes neuronales convolucionales (CNN) y un módulo de visión por computadora para detectar manzanas defectuosas en una máquina clasificadora de cuatro líneas a una velocidad de 5 frutas s^{-1}; además, compararon la precisión de la CNN con un método de procesamiento de imágenes basado en el recuento de regiones defectuosas y con un clasificador de máquina de soporte vectorial, la CNN implementada en la máquina clasificadora obtuvo los mejores resultados.

Los métodos que generalmente se utilizan para la clasificación de variedades y detección de defectos superficiales de la manzana se basan en técnicas de procesamiento de imágenes digitales que consideran la eliminación del fondo, segmentación de defectos y la identificación de las zonas del tallo y del cáliz (Wang *et al*., 2022).

Al respecto, Baneh *et al*. (2018) desarrollaron un sistema electrónico capaz de clasificar manzanas sobre una banda transportadora e informaron de un clasificador neuronal para Golden y Red Delicious que distingue cuatro categorías, de acuerdo con la norma europea, obteniendo una exactitud de 89 % para Golden Delicious y de 92 % para Red Delicious, respectivamente. Wu *et al*. (2020) usaron un modelo de CNN AlexNet modificado con una estructura de 11 capas para identificar y detectar defectos en las manzanas, utilizaron imágenes de retrodispersión inducidas por láser y alcanzaron una exactitud de 92.5 %. Los instrumentos de clasificación mejoran los tiempos de inspección sanitaria, Bhargava y Bansal (2021) afirmaron que la clasificación simultánea de frutas por tamaño y color ahorraría el tiempo de inspección sanitaria, reduciendo significativamente el manejo de la fruta.

Sun *et al*. (2017) consideran que un sistema de clasificación automática de manzanas debe involucrar los aspectos de color, peso, dimensiones y defectos. Los autores destacaron que el brillo superficial de una manzana refleja la frescura y sus defectos, y consideraron a esta característica sensorial entre las más importantes. En el estudio de Sofu *et al*. (2016) se encontró que la característica que más afecta la calidad en la clasificación de la manzana es la mancha y la descomposición.

La inteligencia artificial (IA) es una herramienta que ha tomado relevancia en los últimos años; en ese contexto, el trabajo de Moallem *et al*. (2017) es un buen referente en cuanto a técnicas de clasificación de manzanas utilizando IA, pues reportaron la comparación de diferentes modelos de aprendizaje automático, extrajeron características estadísticas, texturales y geométricas de imágenes de manzanas Golden Delicious y usaron sistemas de clasificación tipo SVM (Máquinas de soporte vectorial), MLP (Perceptrón multicapa) y KNN (K-vecinos cercanos) clasificando las manzanas en saludables y dañadas con una precisión de 92.5 %.

Dentro de la IA están las técnicas de aprendizaje profundo, particularmente las redes neuronales convolucionales (CNN, del inglés *convolutional neuronal networks*), que aplican múltiples capas de filtros convolucionales de una o más dimensiones a una entrada para la extracción de características, las cuales, conectadas a una red perceptrón multicapa pueden realizar tareas de clasificación; las entradas generalmente son imágenes.

Las CNN tienen muchas aplicaciones en varios campos del conocimiento, incluida la agricultura (Kamilaris *et al*., 2018). Se han utilizado con éxito para detectar los defectos en melocotones (*Prunus persica*) (Sun *et al*., 2019) y en pepinos (*Cucumis sativus*) (Liu *et al*., 2018); se informa incluso que se midieron con éxito los residuos de plaguicidas en manzana en poscosecha (Jiang *et al*., 2019), así como enfermedades en las plantas (Barbedo, 2019) y porcentaje de floración en algodón (*Gossypium hirsutum*) (Xu *et al*., 2018). En el trabajo de Fan *et al*. (2020) se reporta el entrenamiento de una CNN para clasificar manzanas de la variedad Gala en sanas y dañadas, implementando su algoritmo en una banda transportadora y obteniendo una exactitud del 92 %.

Para resolver la tarea de clasificación de la manzana, en muchos casos los productores implementan líneas de inspección visual donde personas son entrenadas para identificar clase, tamaño, grado de madurez, textura y daños principalmente; sin embargo, estos métodos son subjetivos y dificultan la inspección de grandes lotes del fruto o análisis en masa, pues son de alto costo y baja eficiencia (Wang *et al*., 2022).

Una propiedad particular de las CNN es que aprenden características de la imagen, pues cada capa convolucional extrae patrones locales en pequeñas ventanas de dos dimensiones (kernel) orientadas a detectar características o rasgos visuales como aristas, líneas, texturas, color, entre otras; además, pueden extraer información de los defectos del fruto en función de las imágenes con las que son entrenadas. Debido a estas cualidades, las CNN fueron consideradas para el desarrollo de esta investigación.

Se define como manzana sana a aquella que cumple con la definición de fruto sano de acuerdo con la norma oficial mexicana NMX-FF-061-SCFI-2003 (SE, 2003), y es la siguiente: "Fruta libre de enfermedades, heridas, pudriciones, daños producidos por insectos u otras plagas,

libre de insectos vivos o muertos o sus larvas". Por otro lado, se define como manzana dañada a aquella cuyas modificaciones son reconocidas en la norma oficial como daño mecánico, por magulladura, picadura, raspadura o herida (SE, 2003).

En la revisión del trabajo relacionado no se encontró información sobre cuál es la red neuronal que mejor clasifica la calidad de las manzanas; por lo tanto, el objetivo de este trabajo fue determinar una arquitectura de red neuronal convolucional (CNN) que permita clasificar manzanas en sanas y dañadas en la etapa de postcosecha. Para encontrar una arquitectura adecuada se tomaron las redes neuronales que reportan mejores resultados y se probaron en la tarea objetivo; a su vez, diversos hiperparámetros se probaron con cada arquitectura.

MATERIALES Y MÉTODOS

Arquitecturas CNN analizadas

Para lograr la clasificación con métodos no invasivos de las cuatro variedades de manzana en categorías sanas y con daños, como primer paso se propuso desarrollar un sistema de visión en sitio con una arquitectura CNN capaz de realizar esta tarea, donde las arquitecturas analizadas fueron LeNet5, que está basada en la arquitectura propuesta por Lecun *et al*. (1998) y VGG16, propuesta por Simonyan y Zisserman (2014). En VGG16 se utilizaron, además, técnicas de aprendizaje por transferencia, específicamente extracción de características (feature extraction) y ajuste fino (fine-tuning), además de entrenamiento desde cero (from scratch). En el aprendizaje por transferencia se utilizó solamente la arquitectura VGG16 que fue pre-entrenada con ImageNet (Russakovsky *et al*., 2015).

Descripción del conjunto de datos

Una de las tareas fue la creación de la base de datos de imágenes de las cuatro categorías de manzanas sanas y con daños de las variedades Gala, Golden Delicious, Granny Smith y Red Delicious. Para capturar las imágenes se diseñó y construyó una banda transportadora (Figura 1), ésta contaba con un espacio aislado de condiciones de iluminación externa, dicho espacio cerrado de acero inoxidable tuvo iluminación controlada compuesta por tiras LED que entregaban 300 lúmenes por metro. El área de captura fue de 200 cm^2 y la altura a la que se encontraban la cámara y la iluminación LED fue de 30 cm. Se utilizó el software libre OpenCV y una cámara (Logitech modelo C920, Newark, California, EUA) con una resolución máxima de 1080p/30fps.

La base de datos de imágenes RGB generada fue de 4800 imágenes con dimensión de 800 × 600 píxeles cada una, que está compuesta por las cuatro variedades de manzanas, organizadas en sanas y dañadas. Los daños considerados para esta base de datos son de origen mecánico, por magulladuras, picaduras, raspaduras y heridas. El 70 % de las imágenes se utilizó para el entrenamiento de las CNN, el 15 % para validación y el 15 % para prueba. Se reorganizaron las 4800 imágenes de manzanas en ocho clases: 600 imágenes de manzana Gala sana (Figura 2A), 600 imágenes de manzana Gala con daños (Figura 2E), 600 imágenes de manzana Golden Delicious sana (Figura 2B), 600 imágenes de manzana Golden Delicious con daños (Figura 2F), 600 imágenes de manzana Granny Smith sana (Figura 2C), 600 imágenes de manzana Granny Smith con daños (Figura 2G), 600 imágenes de manzana Red Delicious sana (Figura 2D) y 600 imágenes de manzana Red Delicious con daños (Figura 2H).

Arquitecturas LeNet5 y VGG16

La arquitectura de red neuronal convolucional LeNet5 de Yann Lecun (Lecun *et al*., 1998) fue diseñada para el conjunto de datos MNIST del problema de reconocimiento de caracteres escritos a mano, logrando una exactitud en clasificación de 99.2 %. Esta arquitectura consta de dos capas convolucionales, dos capas de cribado máximo (max pooling) y tres capas densas (dense). Para el presente estudio se propuso una modificación a las capas densas de la arquitectura LeNet5 de la siguiente manera: a la capa densa1 se le asignaron 256 neuronas con función de activación Relu, a la capa densa2 se le asignaron 84 neuronas con función de activación Relu, y a la capa densa3 se le asignaron ocho neuronas con función de activación SoftMax para poder clasificar las ocho clases (Figura 3).

La red neuronal convolucional VGG16 fue propuesta por Simonyan y Zisserman (2014); esta arquitectura participó en el ILSVRC-2014 (ImageNet Large-Scale Visual Recognition Challenge 2014) y alcanzó una precisión del 92.7 % quedando entre los cinco primeros en ImageNet (Russakovsky *et al*., 2015). La arquitectura VGG16 contó con cinco bloques convolucionales (B1, ..., B5), cada bloque convolucional fue seguido por una capa de agrupación y, finalmente, se tuvieron tres capas densas. En este caso se acondicionaron las capas densas de la arquitectura VGG16 de la siguiente manera: a la capa densa1 se le asignaron 4096 neuronas con función de activación Relu, a la capa densa2 se le asignaron 512 neuronas con función de activación Relu y a la capa densa3 se le asignaron ocho neuronas con función de activación Softmax para poder clasificar las ocho clases, como se muestra en la Figura 3.

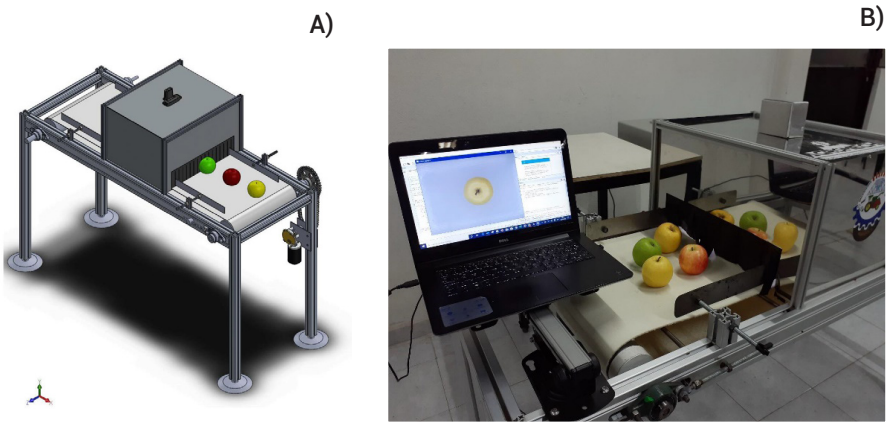


Figura 1. Banda transportadora para captura y clasificación de manzanas. A) diseño de la banda transportadora, B) captura de imágenes de manzanas.

Figura 2. Ejemplos de la base de datos de imágenes de manzanas sanas y con daños.

Diseño del experimento

Se generó una serie de tratamientos (combinación de red con hiperparámetros) que se utilizaron en la clasificación. Al probarse cada tratamiento se midió su rendimiento. Al finalizar el experimento, el tratamiento con mejor rendimiento fue el seleccionado. Los factores involucrados en cada tratamiento fueron la arquitectura, el número de capas a entrenar, la inicialización de los pesos, el optimizador y la tasa de aprendizaje. Debido a que la cantidad de combinaciones es muy alta (aunque son sólo cinco factores, la combinación de los posibles valores en cada factor implica una cantidad de experimentos igual a la combinatoria sin repetición), se decidió realizar el diseño de experimento en dos etapas. En la primera etapa se realizó una búsqueda "rápida" entrenando sólo 30 épocas. Esta primera etapa produce las posibles tasas de aprendizaje. La segunda etapa probó menos combinaciones, pero con más épocas (500). A continuación se describe en detalle cada una de las etapas.

Búsqueda rápida por rejilla

El objetivo de esta fase fue encontrar un conjunto discreto de valores para la tasa de aprendizaje por el método de búsqueda por cuadrícula (Grid search). Este método divide un espacio de búsqueda en intervalos discretos uniformes; en específico, se implementó con el paquete Scikit-learn de Python (Varoquaux *et al*. 2015), el cual cuenta con una herramienta denominada RandomizedSearchCV que puede muestrear una cantidad determinada de candidatos de un espacio de parámetros y permitió delimitar el espacio de búsqueda.

Los aspectos de ajuste de los hiperparámetros fueron la tasa de aprendizaje, que varió desde 1×10^{-8} hasta 0.1 con pasos de 0.001, gamma de 0.001 a 0.0001, kernel de 3, y los optimizadores fueron Adam y RMSProp. Se emplearon las arquitecturas LeNet5 y VGG16 con las modificaciones descritas en la Figura 3; éstas fueron entrenadas desde cero y se estableció un número de 30 épocas para cada búsqueda. En esta primera instancia la búsqueda entregó como mejores tasas de aprendizaje las siguientes: 1×10^{-3}, 1×10^{-4}, 1×10^{-5}, 1.5×10^{-5} y 1×10^{-6}.

Búsqueda completa por rejilla

El objetivo de esta fase fue determinar la arquitectura y los hiperparámetros que mejor se desempeñan en esta tarea. A cada combinación de arquitectura con hiperparámetros se le nombró tratamiento. En el Cuadro 1 se muestran los tratamientos que se probaron. Las arquitecturas fueron LeNet5 y VGG16. Los optimizadores fueron Adam y RMSProp. Las tasas de aprendizaje fueron las cinco que se incluyeron en el experimento anterior. Las técnicas de entrenamiento fueron tres: entrenamiento desde cero, por extracción de características y ajuste fino. El entrenamiento desde cero inicializa de forma aleatoria todos los pesos de la CNN. La extracción de características y el ajuste fino sólo aplica para VGG16. La extracción de características primero inicializa los pesos con una red pre entrenada, segundo, fija los pesos del bloque de extracción de características, y tercero, habilita para su modificación a las capas densas. El ajuste fino primero inicializa los pesos con una red pre entrenada, y posteriormente fija los pesos de los bloques 1 al 4, habilita el bloque 5 y las capas densas para su modificación, lo que se observa en las capas de la red en las Figuras 3B y 3C.

En el experimento se realizó un preprocesamiento de imagen que implica un re-escalamiento de cada imagen a 256 × 256 píxeles y una normalización (Figura 3A). Se estableció para el entrenamiento un tamaño de lote por época (Batch size) de 90 imágenes. Los entrenamientos fueron de 500 épocas con una espera (patience) de 10 épocas para detener el entrenamiento si la perdida no disminuye.

La implementación se realizó con software libre, específicamente con Python 3.6, OpenCV 4.1.2, Tensorflow 2.6.0, Keras y Scikit-learn (Varoquaux *et al*., 2015). En el caso del hardware, se usó la plataforma Google Colaboratory.

Métricas de evaluación

Para evaluar el desempeño de cada tratamiento se utilizaron las métricas de exactitud, recuerdo, puntuación F1, precisión y matriz de confusión, las cuales son definidas en la literatura (Raschka y Mirjalili, 2019).

RESULTADOS

Los resultados experimentales de la búsqueda completa por rejilla se presentan en los Cuadros 2 al 5. En el Cuadro 2 se puede observar que la exactitud en el conjunto de prueba se encuentra en el intervalo de 0.61 a 0.97, y que en los tratamientos (1 al 10) la arquitectura LeNet5 con optimizador RMSProp y tasa de aprendizaje 1×10^{-4} es la de mejor desempeño, con una exactitud de 0.97. En el Cuadro 3 se puede observar que la exactitud en el conjunto de prueba para los tratamientos (11 al 20) se encuentra en el intervalo de 0.12 a 0.95, y que la arquitectura VGG16 con optimizador Adam y tasa de aprendizaje 1×10^{-5} es la de mejor desempeño, con una exactitud de 0.95. El Cuadro 4, que corresponde a los tratamientos 21 al 30, muestra una exactitud en el conjunto de prueba con un intervalo de 0.91 a 0.95, y la arquitectura VGG16 con optimizador Adam, tasa de aprendizaje de 1×10^{-5} y con extracción

Cuadro 1. Cuarenta experimentos para medir rendimiento en clasificación con los tratamientos evaluados.

Identificador	T. E.	Arquitectura	Optimizador	T. A.
1-5	Desde cero	LeNet5	RMSProp	5 posibles†
6-10	Desde cero	LeNet5	Adam	5 posibles†
11-15	Desde cero	VGG16	RMSProp	5 posibles†
16-20	Desde cero	VGG16	Adam	5 posibles†
21-25	Extracción de características	VGG16	RMSProp	5 posibles†
26-30	Extracción de características	VGG16	Adam	5 posibles†
31-35	Ajuste fino	VGG16	RMSProp	5 posibles†
36-40	Ajuste fino	VGG16	Adam	5 posibles†

T. E.: Técnica de entrenamiento, T. A.: tasa de aprendizaje. †Cada renglón condensa cinco unidades experimentales donde se usa una de cinco posibles tasas de aprendizaje: 1×10^{-3}, 1×10^{-4}, 1×10^{-5}, 1.5×10^{-5} y 1×10^{-6}

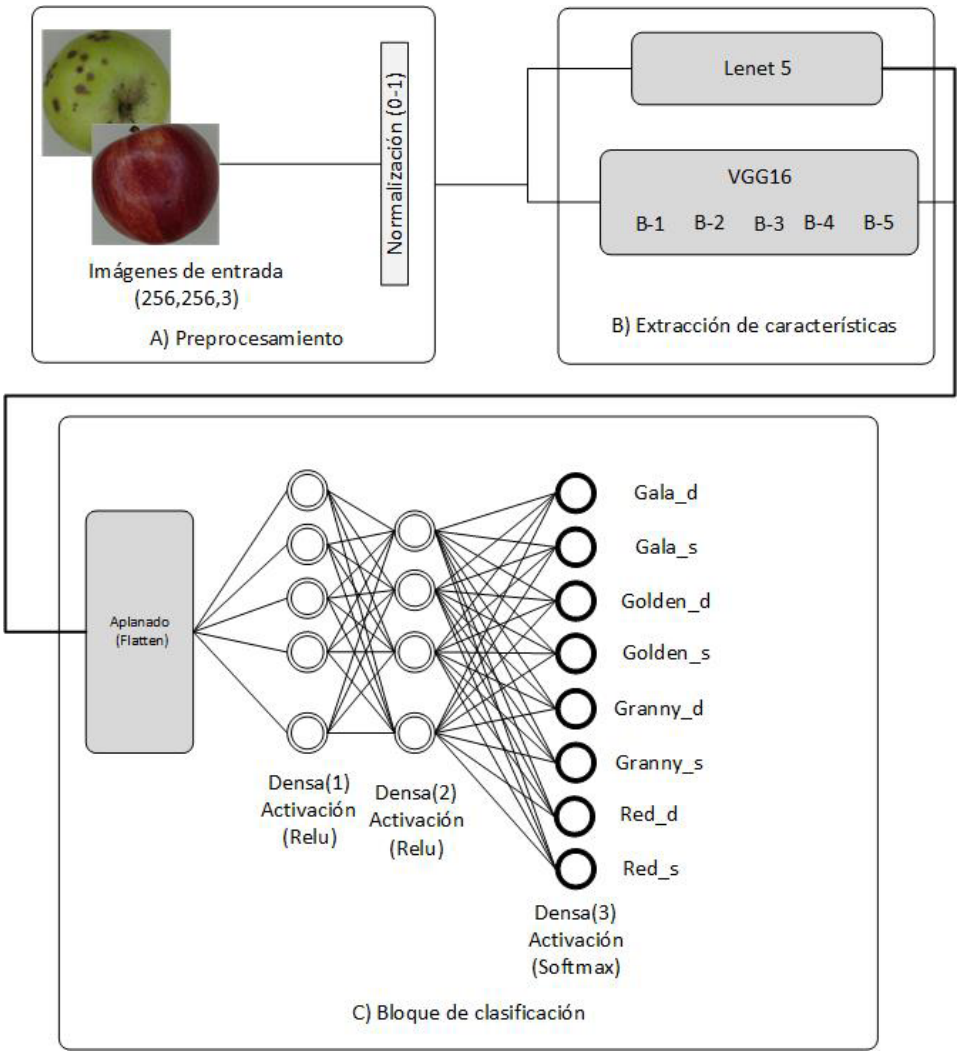


Figura 3. Modelos de las arquitecturas LeNet5 y VGG16 utilizadas para la extracción de característica y clasificación de manzanas.

Cuadro 2. Rendimiento en clasificación de los tratamientos con LeNet5 entrenada desde cero.

Hiperparámetros de entrenamiento				Rendimiento en clasificación			
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud validación	Exactitud prueba
1	RMSProp	1.0×10^{-3}	Random	21	0.90	0.66	0.61
2	RMSProp	1.0×10^{-4}	Random	44	0.99	0.98	0.97
3	RMSProp	1.0×10^{-5}	Random	174	0.98	0.94	0.94
4	RMSProp	1.5×10^{-5}	Random	131	0.96	0.94	0.92
5	RMSProp	1.0×10^{-6}	Random	226	0.94	0.88	0.87
6	Adam	1.0×10^{-3}	Random	15	0.70	0.58	0.62
7	Adam	1.0×10^{-4}	Random	36	0.99	0.96	0.95
8	Adam	1.0×10^{-5}	Random	65	0.96	0.90	0.89
9	Adam	1.5×10^{-5}	Random	64	0.97	0.94	0.93
10	Adam	1.0×10^{-6}	Random	247	0.95	0.91	0.92

Trat: Tratamiento.

Cuadro 3. Rendimiento en clasificación de los tratamientos con VGG16 entrenada desde cero.

Hiperparámetros de entrenamiento				Rendimiento en clasificación			
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud validación	Exactitud prueba
11	RMSProp	1.0×10^{-3}	Random	12	0.10	0.12	0.12
12	RMSProp	1.0×10^{-4}	Random	41	0.98	0.94	0.83
13	RMSProp	1.0×10^{-5}	Random	103	0.98	0.95	0.94
14	RMSProp	1.5×10^{-5}	Random	74	0.97	0.96	0.95
15	RMSProp	1.0×10^{-6}	Random	111	0.65	0.57	0.58
16	Adam	1.0×10^{-3}	Random	17	0.11	0.12	0.12
17	Adam	1.0×10^{-4}	Random	46	0.99	0.87	0.84
18	Adam	1.0×10^{-5}	Random	71	0.99	0.97	0.95
19	Adam	1.5×10^{-5}	Random	91	0.99	0.94	0.88
20	Adam	1.0×10^{-6}	Random	99	0.77	0.73	0.66

Trat: Tratamiento.

de características tuvo exactitud de 0.95. El Cuadro 5, que corresponde a los tratamientos 31 al 40 muestra una exactitud en el conjunto de prueba con un intervalo entre 0.94 y 0.96, y la arquitectura VGG16 con optimizador Adam, tasa de aprendizaje de 1.5×10^{-5} y ajuste fino tuvo exactitud de 0.96.

Se determinó que LeNet5, con el tratamiento 2, fue la arquitectura de mejor desempeño, pues obtuvo una exactitud de 97 %; por lo tanto, es la mejor arquitectura.

En el Cuadro 6 se reportan las métricas de precisión, sensibilidad y F1 de la arquitectura ganadora. En la evaluación de la matriz de confusión, se utilizaron 720 imágenes y el desempeño en clasificación se muestra en la Figura 4A sin normalizar y en la Figura 4B normalizada.

DISCUSION

LeNet5 con el tratamiento 2 clasifica bien (100% en la matriz de confusión) las categorías Gala sana, Golden Delicious dañada y Granny sana; para las cinco categorías

Cuadro 4. Rendimiento en clasificación de los tratamientos con VGG16 y extracción de características.

Hiperparámetros de entrenamiento				Rendimiento en clasificación			
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
21	RMSProp	1.0×10^{-3}	ImageNet	36	0.97	0.94	0.93
22	RMSProp	1.0×10^{-4}	ImageNet	28	0.98	0.93	0.91
23	RMSProp	1.0×10^{-5}	ImageNet	90	0.99	0.95	0.94
24	RMSProp	1.5×10^{-5}	ImageNet	51	1.00	0.95	0.94
25	RMSProp	1.0×10^{-6}	ImageNet	190	1.00	0.95	0.95
26	Adam	1.0×10^{-3}	ImageNet	22	1.00	0.95	0.92
27	Adam	1.0×10^{-4}	ImageNet	55	1.00	0.95	0.95
28	Adam	1.0×10^{-5}	ImageNet	83	1.00	0.96	0.95
29	Adam	1.5×10^{-5}	ImageNet	66	1.00	0.95	0.95
30	Adam	1.0×10^{-6}	ImageNet	267	1.00	0.95	0.95

Trat: Tratamiento.

Cuadro 5. Rendimiento en clasificación de los tratamientos con VGG16 y ajuste fino.

Hiperparámetros de entrenamiento				Rendimiento en clasificación			
Trat.	Optimizador	Tasa de aprendizaje	Pesos iniciales	Épocas	Exactitud entrenamiento	Exactitud Validación	Exactitud prueba
31	RMSProp	1.0×10^{-3}	ImageNet	23	0.96	0.78	0.94
32	RMSProp	1.0×10^{-4}	ImageNet	25	0.98	0.94	0.96
33	RMSProp	1.0×10^{-5}	ImageNet	60	1.00	0.96	0.95
34	RMSProp	1.5×10^{-5}	ImageNet	51	1.00	0.97	0.95
35	RMSProp	1.0×10^{-6}	ImageNet	120	1.00	0.96	0.95
36	Adam	1.0×10^{-3}	ImageNet	27	1.00	0.97	0.96
37	Adam	1.0×10^{-4}	ImageNet	36	1.00	0.98	0.95
38	Adam	1.0×10^{-5}	ImageNet	65	1.00	0.97	0.96
39	Adam	1.5×10^{-5}	ImageNet	70	1.00	0.97	0.96
40	Adam	1.0×10^{-6}	ImageNet	178	1.00	0.96	0.95

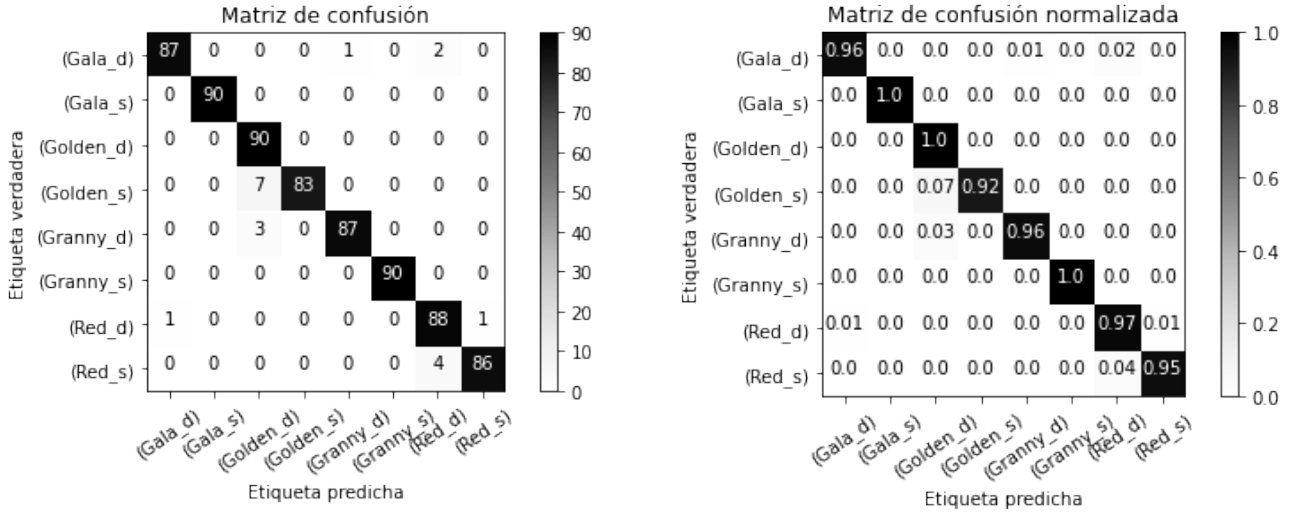
Trat: Tratamiento.

de manzanas restantes, los porcentajes se encontraron entre 92 y 97% (Figura 4B). Fan *et al*. (2020) informaron que su arquitectura de CNN entrenada desde cero para clasificar manzanas sanas y dañadas de la variedad Gala obtuvo 92 % de exactitud con un conjunto de 200 imágenes para la prueba; además, con técnicas clásicas de visión artificial (Sofu *et al*., 2016) reportaron que utilizaron variedades de manzana Granny Smith y Golden Delicious para clasificar sólo manzanas sanas, y obtuvieron una exactitud del 89 %. Kayaalp y Metlek (2020) clasificaron manzanas Gala sanas y dañadas usando clasificadores de arquitectura profunda (CNN) y reportaron tasas de exactitud en clasificación de 91.25 %; en contraste, Moallem *et al*. (2017) reportaron que para realizar la clasificación utilizaron 16 manzanas Golden sanas y ocho manzanas Golden dañadas y obtuvieron en cada clase un desempeño del 92.5 %. Al analizar los resultados obtenidos de la matriz de confusión normalizada (Figura 4B) se observa también que la manzana Red Delicious sana se clasifica con un 95 % de exactitud y la Red Delicious dañada con 97 %; además, se muestra que el desempeño en clasificación para la manzana Gala sana es de 100 % y para la manzana

Cuadro 6. Métricas de desempeño en clasificación de LeNet5 con el tratamiento 2.

Categoría	Precisión	Sensibilidad	Puntuación F1	No. de imágenes
Gala_d	0.99	0.97	0.98	90
Gala_s	1.00	1.00	1.00	90
Golden_d	0.90	1.00	0.95	90
Golden_s	1.00	0.92	0.96	90
Granny_d	0.99	0.97	0.98	90
Granny_s	1.00	1.00	1.00	90
Red_d	0.94	0.98	0.96	90
Red_s	0.99	0.96	0.97	90

_d: dañada, _s: sana.



A) Matriz de confusión de LeNet5 con el tratamiento 2

B) Matriz de confusión normalizada de LeNet5 con el tratamiento 2

Figura 4. Matriz de confusión de la arquitectura ganadora, _s: sana, _d: dañada.

Gala dañada de 96 %. Fan *et al*. (2020) reportaron que en su matriz de confusión se alcanzó un 93 % de exactitud para la clase Gala sana y 91 % para Gala dañada. En la literatura se informa que Sofu *et al*. (2016) obtuvieron un desempeño en clasificación del 93.4 % para la categoría Granny Smith sana y 96 % para la clase dañada.

Finalmente, se hace notar que el tamaño de arquitectura de LeNet5 es casi tres veces menor en comparación con VGG16, ya que el número de parámetros (pesos de la red), considerando entrenamientos desde cero, es de 47,298,476 y 151,038,280 respectivamente, ello implica que para un conjunto de imágenes como el presentado en este estudio de 4800 organizadas en ocho clases, una arquitectura convolucional como LeNet5 entrenada desde cero es adecuada para cumplir con esta tarea.

CONCLUSIONES

Se determinó que la red neuronal LeNet5 entrenada desde cero, con optimizador RMSProp y tasa de aprendizaje 1 × 10^{-4} logra clasificar correctamente manzanas sanas y dañadas de las variedades Red Delicious, Granny Smith, Golden Delicious y Gala, con una exactitud del 97 %; sin embargo, considerando la matriz de confusión, se puede inferir que esta arquitectura funciona al 100 % para clasificar las categorías Gala sana, Golden dañada y Granny Smith sana; para las cinco categorías restantes, el porcentaje varía entre 92 y 97 %, lo cual es importante porque la norma oficial mexicana (SE, 2003) permite que hasta 10 % de las manzanas de cualquier lote no reúna los requisitos enunciados. Para fines de una implementación que considere cumplir con la norma oficial mexicana NMX-FF-061-SCFI-2003 la arquitectura encontrada funciona. Un hallazgo en este trabajo fue que el diseño experimental

generado permite encontrar específicamente los hiperparámetros y la arquitectura con el mejor desempeño en términos de clasificación. Debido a que la búsqueda rápida por rejilla se limitó a un espacio menor de posibles valores de hiperparámetros, en la búsqueda completa por rejilla se realizaron sólo 40 combinaciones para determinar la mejor arquitectura; es decir, con esta técnica se ajustó gradualmente el espacio de búsqueda a un lote más pequeño de posibles combinaciones.

BIBLIOGRAFÍA

Baneh N. M., H. Navid and J. Kafashan (2018) Mechatronic components in apple sorting machines with computer vision. *Journal of Food Measurement and Characterization* 12:1135-1155, https://doi.org/10.1007/s11694-018-9728-1

Barbedo J. G. A. (2019) Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering* 180:96-107, http://doi.org/10.1016/j.biosystemseng.2019.02.002

Bhargava A. and A. Bansal (2021) Fruits and vegetables quality evaluation using computer vision: a review. *Journal of King Saud University - Computer and Information Sciences* 33:243-257, https://doi.org/10.1016/j.jksuci.2018.06.002

Fan S., J. Li, Y. Zhang, X. Tian, Q. Wang, X. He, … and W. Huang (2020) Online detection of defective apples using computer vision system combined with deep learning methods. *Journal of Food Engineering* 286:110102, https://doi.org/10.1016/j.jfoodeng.2020.110102

FAO, Organización de las Naciones Unidas para la Alimentación y la Agricultura (2010) Norma para las manzanas (CODEX STAN 299-2010). Codex Alimentarius: Normas Internacionales de los Alimentos. Washington, D. C., USA. https://www.fao.org/fao-who-codexalimentarius/codex-texts/list-standards/es/ (Octubre 2021).

Jiang B., J. He, S. Yang, H. Fu, T. Li, H. Song and D. He (2019) Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. *Artificial Intelligence in Agriculture* 1:1-8, https://doi.org/10.1016/j.aiia.2019.02.001

Kamilaris A. and F. X. Prenafeta-Boldú (2018) Deep learning in agriculture: a survey. *Computers and Electronics in Agriculture* 147:70-90, https://doi.org/10.1016/j.compag.2018.02.016

Kayaalp K. and S. Metlek (2020) Classification of robust and rotten apples by deep learning algorithm. *Sakarya University Journal of Computer and Information Sciences* 3:112-120, http://doi.org/10.35377/saucis.03.02.717452

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

Liu Z., Y. He, H. Cen and R. Lu (2018) Deep feature representation with stacked sparse auto-encoder and convolutional neural network for hyperspectral imaging-based detection of cucumber defects. *Transactions of the ASABE* 61:425-436, https://doi.org/10.13031/trans.12214

Lu Y. and R. Lu (2017) Non-destructive defect detection of apples by spectroscopic and imaging technologies: a review. *Transactions of the ASABE* 60:1765-1790, https://doi.org/10.13031/trans.12431

Moallem P., A. Serajoddin and H. Pourghassem (2017) Computer vision-based apple grading for golden delicious apples based on surface features. *Information Processing in Agriculture* 4:33-40, https://doi.org/10.1016/j.inpa.2016.10.003

Raschka S. and V. Mirjalili (2019) Python Machine Learning. Segunda edición. Marcombo Ediciones Técnicas. Barcelona, España. 618 p.

Russakovsky O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, ... and L. Fei-Fei (2015) ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115:211-252, https://doi.org/10.1007/s11263-015-0816-y

SE, Secretaría de Economía (2003) NMX-FF-061-SCFI-2003. Productos agrícolas no industrializados para consumo humano - fruta fresca- manzana (*Malus pumila* Mill) - (*Malus domestica* Borkh) - especificaciones. http://intranet.dif.cdmx.gob.mx/transparencia/new/art_121/52/_anexos/normamexicanaproductosagricolasnoindustrializados.pdf (Agosto 2022).

SIAP, Servicio de Información Agropecuaria y Pesquera (2020) Panorama agroalimentario 2020. Secretaría de Agricultura y Desarrollo Rural. Ciudad de México. https://nube.siap.gob.mx/gobmx_publicaciones_siap/pag/2020/Atlas-Agroalimentario-2020 (Octubre 2021).

Simonyan K. and A. Zisserman (2014) Very deep convolutional networks for large-scale image recognition. *In*: 3rd International Conference on Learning Representations. San Diego, California, 7-9 May. Y. Bengio and Y. Lecun (eds.). Conference Track Proceedings. Cornell University. Ithaca, New York, USA. pp:1-14, https://doi.org/10.48550/arXiv.1409.1556

Sofu M. M., O. Er, M. C. Kayacan and B. Cetişli (2016) Design of an automatic apple sorting system using machine vision. *Computers and Electronics in Agriculture* 127:395-405, https://doi.org/10.1016/j.compag.2016.06.030

Sun K., Y. Li, J. Peng, K. Tu and L. Pan (2017) Surface gloss evaluation of apples based on computer vision and support vector machine method. *Food Analytical Methods* 10:2800-2806, https://doi.org/10.1007/s12161-017-0849-7

Sun Y., R. Lu, Y. Lu, K. Tu and L. Pan (2019) Detection of early decay in peaches by structured-illumination reflectance imaging. *Postharvest Biology and Technology* 151:68-78, https://doi.org/10.1016/j.postharvbio.2019.01.011

Varoquaux G., L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa and A. Mueller (2015) Scikit-learn: machine learning without learning the machinery. *GetMobile: Mobile Computing and Communications* 19:29-33, https://doi.org/10.1145/2786984.2786995

Wang Z., L. Jin, S. Wang and H. Xu (2022) Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. *Postharvest Biology and Technology* 185:111808, https://doi.org/10.1016/j.postharvbio.2021.111808

Wu A., J. Zhu and T. Ren (2020) Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Computers & Electrical Engineering* 81:106454, https://doi.org/10.1016/j.compeleceng.2019.106454

Xu R., C. Li, A. H. Paterson, Y. Jiang, S. Sun and J. S. Robertson (2018) Aerial images and convolutional neural network for cotton bloom detection. *Frontiers in Plant Science* 8:2235, https://doi.org/10.3389/fpls.2017.02235

Zhang Z., A. K. Pothula and R. Lu (2017) Economic evaluation of apple harvest and in-field sorting technology. *Transactions of the ASABE* 60:1537-1550, https://doi.org/10.13031/trans.12226