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.

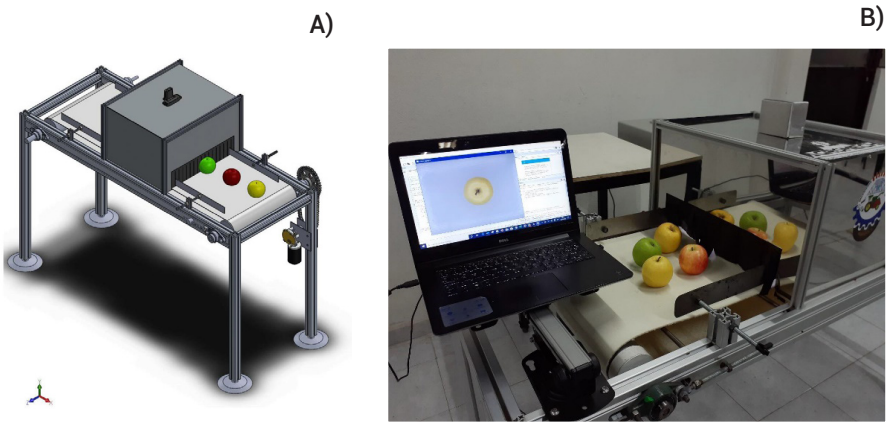


Figura 1. Banda transportadora para captura y clasificación de manzanas. A) diseño de la banda transportadora, B) captura de imágenes de manzanas.

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}

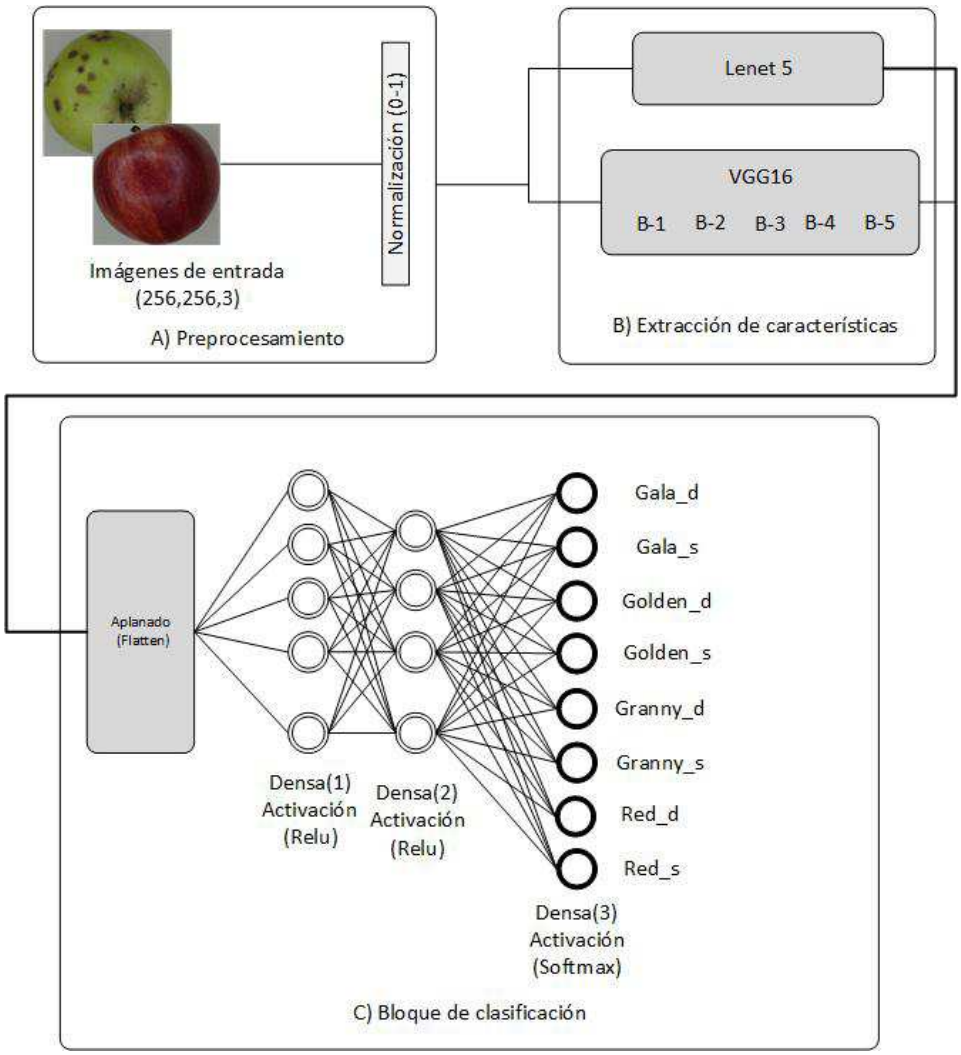


Figura 3. Modelos de las arquitecturas LeNet5 y VGG16 utilizadas para la extracción de característica y clasificación de manzanas.

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

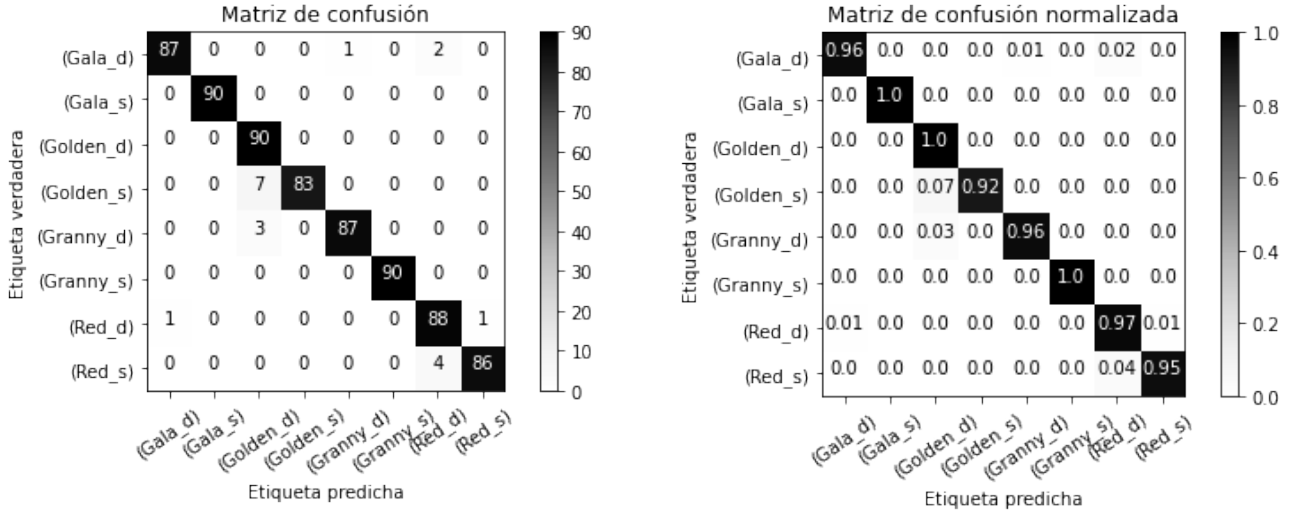
Trat: Tratamiento.

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