

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Detección de defectos en frutas usando modelos de CNN con datos
reales y virtuales

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero/a en Ciencias de la Computación

Presentado por:

Paula Cristina González Freire

Renzo Stalin Pacheco Jiménez

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto se lo dedico principalmente a Dios, por ser mi principal motor desde que tengo memoria, a mi abuelita, quien me ha prestado sus fuerzas y dedicación desde que nací y quien siempre soñó poder estar presente en mi graduación, a mis padres por su apoyo incondicional, a mi mano derecha Guillermo Bernal, quien ha sido parte fundamental durante el transcurso de mi carrera, por su apoyo emocional, de conocimiento y tecnológico.

Paula Cristina González Freire

DEDICATORIA

El presente proyecto se lo dedico a mis padres, mis hermanos y el resto de mi familia, quienes son un pilar fundamental en mi vida, sin su apoyo constante e incondicional no hubiese podido alcanzar las metas personales, académicas y profesionales que me he propuesto a lo largo de los años.

Renzo Stalin Pacheco Jiménez

AGRADECIMIENTOS

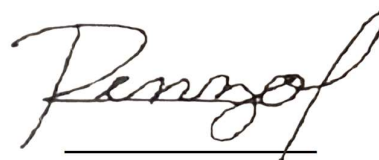
Nuestro más sincero agradecimiento al PhD. Boris Vintimilla, quien como tutor nos guió y aportó con sus conocimientos en la realización de este proyecto y al MSc. Luis Chuquimarca, quien estuvo siempre dispuesto a colaborar con recursos y conocimientos a lo largo de la elaboración del presente proyecto.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Paula Cristina González Freire y Renzo Stalin Pacheco Jiménez, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

A handwritten signature in black ink, reading "Paula González Freire". The signature is written in a cursive style with a period at the end. A horizontal line is drawn underneath the signature.

Paula Cristina González Freire

A handwritten signature in black ink, reading "Renzo Pacheco Jiménez". The signature is written in a cursive style. A horizontal line is drawn underneath the signature.

Renzo Stalin Pacheco Jiménez

EVALUADORES

PhD. Boris Vintimilla

PROFESOR DE LA MATERIA

PhD. Boris Vintimilla

PROFESOR TUTOR

RESUMEN

El presente proyecto busca evaluar distintos modelos *CNN* con el fin de comparar su rendimiento en el reconocimiento de una gama de defectos en manzanas y mangos para garantizar la calidad de la producción de estos alimentos.

Usando los modelos *CNN*, *InceptionV3*, *MobileNetV2*, *VGG16* y *DenseNet121*, los cuales se entrenaron con un dataset de imágenes reales y sintéticas de manzanas y mangos que abarcan la fruta en condiciones aceptables de calidad y con los defectos: podredumbre, golpes, costras y puntos negros. Los entrenamientos se realizaron con variaciones en los hiper-parámetros para obtener las métricas de rendimiento: *accuracy*, *precision*, *recall* y *F1-score*.

El modelo *MobileNetV2* alcanzó el *accuracy* más alto en los entrenamientos y pruebas, obteniendo 97.90% para manzanas y 95.65% para mangos, siendo este el modelo más apto para la detección de defectos en estas frutas. Los modelos *InceptionV3* y *DenseNet121* presentaron valores de *accuracy* superiores al 90%, mientras que el modelo *VGG16* obtuvo el rendimiento más pobre al no superar el 82% de *accuracy* para ninguna de las frutas.

Los modelos entrenados, sobre todo *MobileNetV2*, son capaces de reconocer una gama de defectos en las frutas objeto de estudio con un alto grado de exactitud y son adecuados para su uso en el desarrollo de aplicaciones de automatización del proceso de evaluación de la calidad de manzanas y mangos.

Palabras clave: Defectos en frutas, Redes neuronales convolucionales, *Transfer learning*

ABSTRACT

This project seeks to evaluate the performance of various CNN models in the identification of defects on apples and mangoes to guarantee the quality of these products.

Using the CNN models InceptionV3, MobileNetV2, VGG16 and DenseNet121, which were trained using a dataset consisting of real and synthetic images of apples and mangoes in acceptable conditions and the defects: rot, bruises, scabs, black spots. For the training of the models the hyper parameters were experimented with to obtain the performance metrics: accuracy, precision, recall and F1-score.

The MobileNetV2 model reached the highest accuracy values during training and testing with 97.90% accuracy for apples and 95.65% accuracy for mangoes, this model is the most apt for defect detection on these fruits. The models InceptionV3 and DenseNet121, also reached accuracy values above 90%, however the model VGG16 had the poorest performance not reaching accuracy values over 82% for any of the fruits.

The trained models, specially MobileNetV2, can recognize a variety of defects on apples and mangoes with a high degree of accuracy and can be used in the development of automation applications for apple and mango quality assessment.

Keywords: *Fruit defects, Convolutional neural networks, Transfer learning*

ÍNDICE GENERAL

RESUMEN	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL	III
ABREVIATURAS	V
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. INTRODUCCIÓN	1
1.1 Descripción de problema	1
1.2 Justificación del problema	2
1.3 Objetivos	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos	2
1.4 Resultados esperados	3
1.5 Impacto comercial	3
1.6 Marco Teórico	3
1.6.1 Defectos en frutas y hortalizas	3
1.6.2 Defectos en manzanas y mangos	5
1.6.3 Técnicas para la detección de defectos en frutas	8
1.6.4 Detección y clasificación de defectos en frutas por medio de visión por computador y modelos <i>CNN</i>	9
1.6.5 Técnicas de adquisición de imágenes	12
CAPÍTULO 2	14
2. METODOLOGÍA	14
2.1 Módulos de desarrollo de la solución	14
2.2 Módulo 1: Creación del dataset	15
2.2.1 Recopilación de imágenes reales y generación de imágenes sintéticas	15
2.1.2 Refinamiento y preprocesamiento de imágenes	18
2.3 Módulo 2: Entrenamiento de modelos de CNN	19
2.3.1 MobileNetV2	20
2.3.2 DenseNet121	21
2.3.3 InceptionV3	22
2.3.4 VGG16	23
2.3.5 Selección de hiper-parámetros	24

2.4 Módulo 3: Evaluación de modelos de CNN	25
2.5 Módulo 4: Desarrollo del prototipo	26
CAPÍTULO 3	27
3. RESULTADOS Y ANÁLISIS	27
3.1 Resultados con el dataset de imágenes de manzanas	27
3.2 Resultados con el dataset de imágenes de mangos	33
CAPÍTULO 4	42
4. CONCLUSIONES Y RECOMENDACIONES	42
4.1 Conclusiones	42
4.2 Recomendaciones	43
4.3 Trabajos futuros	43
4.4 Contribuciones	44
BIBLIOGRAFÍA	46

ABREVIATURAS

BART	<i>Bayesian Additive Regression Trees</i>
CNN	<i>Convolutional Neural Network</i>
GAN	<i>Generative Adversarial Network</i>
LiDAR	<i>Light Detection and Ranging</i>
MCC	<i>Matthew's Correlation Coefficient</i>
MSE	<i>Mean Squared Error</i>
ReLU	<i>Rectified Linear Activation Unit</i>
RGB	<i>Red Green Blue</i>
VGG	<i>Visual Geometry Group</i>
VQGAN	<i>Vector Quantized Generative Adversarial Network</i>

ÍNDICE DE FIGURAS

Figura 1.1 Tipos comunes de defectos en productos hortícolas (traducido al español)	4
Figura 1.2 Manzana en estado de podredumbre	5
Figura 1.3 Mango con golpes	6
Figura 1.4 Manzana con costras	7
Figura 1.5 Puntos negros por infección bacteriana en mangos	7
Figura 2.1 Módulos de desarrollo de la solución	14
Figura 2.2 Estructura del dataset	15
Figura 2.3 Imágenes sintéticas generadas con <i>DALL-E</i> mini para el defecto de podredumbre en manzanas	17
Figura 2.4 Arquitectura de <i>MobileNetV2</i>	20
Figura 2.5 Arquitectura de <i>DenseNet121</i>	21
Figura 2.6 Capas paralelas en el modelo <i>Inception</i> original	22
Figura 2.7 Capas paralelas en el modelo <i>InceptionV3</i>	23
Figura 2.8 Capas del modelo <i>VGG16</i>	24
Figura 2.9 Ecuaciones de las métricas	25
Figura 3.1 <i>Validation accuracy</i> para los entrenamientos con el dataset de imágenes de manzana	30
Figura 3.2 <i>Validation precision</i> para los entrenamientos con el dataset de imágenes de manzana	31
Figura 3.3 <i>Validation recall</i> para los entrenamientos con el dataset de imágenes de manzana	32
Figura 3.4 <i>Validation loss</i> para los entrenamientos con el dataset de imágenes de manzana	33
Figura 3.5 <i>Validation accuracy</i> para los entrenamientos con el dataset de imágenes de mango	36
Figura 3.6 <i>Validation precision</i> para los entrenamientos con el dataset de imágenes de mango	37
Figura 3.7 <i>Validation recall</i> para los entrenamientos con el dataset de imágenes de mango	37
Figura 3.8 <i>Validation loss</i> para los entrenamientos con el dataset de imágenes de mango	38

Figura 3.9 Resultado de MobileNetV2 de manzana con defectos	39
Figura 3.10 Resultado de MobileNetV2 de manzana sin defectos	39
Figura 3.11 Resultado de MobileNetV2 de mango con defectos	40
Figura 3.12 Resultado de MobileNetV2 de mango sin defectos	40

ÍNDICE DE TABLAS

Tabla 1.1 Defectos considerados para la evaluación de modelos <i>CNN</i>	8
Tabla 1.2 Comparación de modelos de <i>CNN</i> en la detección de defectos en manzanas y mangos	11
Tabla 1.3 Modelos <i>CNN</i> escogidos por su desempeño en detección de defectos	12
Tabla 2.1 Fuentes de imágenes reales para creación del dataset	16
Tabla 2.2 Entradas de texto para la generación de imágenes con <i>DALL-E</i> mini	17
Tabla 2.3 Distribución del dataset	19
Tabla 2.4 Hiper-parámetros de entrenamiento	24
Tabla 3.1 Métricas de rendimiento de <i>InceptionV3</i> para el dataset de imágenes de manzanas	28
Tabla 3.2 Métricas de rendimiento de <i>MobileNetV2</i> para el dataset de imágenes de manzanas	28
Tabla 3.3 Métricas de rendimiento de <i>VGG16</i> para el dataset de imágenes de manzanas	29
Tabla 3.4 Métricas de rendimiento de <i>DenseNet121</i> para el dataset de imágenes de manzanas	29
Tabla 3.5 Métricas de rendimiento de <i>InceptionV3</i> para el dataset de imágenes de mangos	34
Tabla 3.6 Métricas de rendimiento de <i>MobileNetV2</i> para el dataset de imágenes de mangos	34
Tabla 3.7 Métricas de rendimiento de <i>VGG16</i> para el dataset de imágenes de mangos	35
Tabla 3.8 Métricas de rendimiento de <i>DenseNet121</i> para el dataset de imágenes de mangos	35

CAPÍTULO 1

1. INTRODUCCIÓN

En este capítulo se presenta la descripción del problema a resolver, así como los objetivos, resultados esperados de la investigación y la revisión del estado del arte respecto al uso de modelos de *CNN* para la detección de defectos en frutas.

1.1 Descripción de problema

La industria alimentaria está sujeta a estrictas normas de calidad que rigen todas las partes del proceso de producción de los alimentos que se consumen. Garantizar la calidad de los alimentos es de suma importancia para la salud y la producción de alimentos de alta calidad supone que estos cumplan con las cualidades aceptables para los consumidores y entes reguladores. Aparte de las cualidades como color, olor, textura, forma y apariencia externa de fácil inspección, también hay cualidades referentes a la higiene e incluso composición química de los alimentos.

Actualmente, se emplean procesos manuales y basados en el criterio de personas para la inspección de la calidad de las frutas y vegetales. Estas técnicas son lentas, imprecisas y dan cabida a la aparición de defectos que provocan que la fruta sea rechazada en los controles de calidad y por los consumidores. Para solucionar esta problemática, se han estudiado técnicas de aprendizaje de máquina y visión por computador que permiten evaluar la calidad de los alimentos de forma más rápida y precisa.

Lograr identificar defectos de forma temprana en las frutas es de importancia para asegurar la calidad de estos alimentos, mantener el valor nutricional de las frutas y la satisfacción del consumidor final, además de evitar pérdidas monetarias para los productores.

El presente proyecto busca realizar una evaluación de modelos de *CNN* para la detección de defectos en manzanas y mangos. La evaluación se realizará usando un conjunto de imágenes reales y sintéticas como datos de entrada para entrenar y validar los modelos. Los defectos que se busca identificar son: podredumbre, golpes, costras y puntos negros, provocados por insectos, enfermedades, condiciones climáticas y la manipulación post cosecha.

1.2 Justificación del problema

Según [1], en países latinoamericanos aproximadamente entre el 10 y 20% de las frutas y hortalizas cosechadas son desechadas por diversos motivos, incluyendo el no cumplimiento de estándares de estética y calidad.

La inspección de la calidad de las frutas luego de la cosecha se realiza de forma manual dando lugar a imprecisiones en la evaluación. El control de calidad por medio de técnicas de visión por computador y aprendizaje de máquina busca disminuir el tiempo de inspección y al mismo tiempo aumentar la precisión en la detección de defectos en las frutas, evitando así las pérdidas económicas del sector agricultor por errores en la producción.

1.3 Objetivos

1.3.1 Objetivo General

Evaluar modelos de *CNN* que permitan la detección de una variedad de defectos en las frutas (manzanas y mangos) para garantizar el cumplimiento de los estándares de calidad para estos alimentos.

1.3.2 Objetivos Específicos

1. Generar un conjunto de imágenes de frutas reales y sintéticas (manzanas y mangos) representativo de una variedad de defectos incluyendo podredumbre, golpes, costras y puntos negros para el entrenamiento y validación de los modelos de *CNN*.
2. Evaluar un conjunto de modelos *CNN* entrenados usando *transfer learning* para identificar una variedad de defectos en frutas (manzanas y mangos).
3. Desarrollar una herramienta *web* que permita la selección y uso de modelos de *CNN* para generar estimaciones sobre la condición de las frutas por medio de imágenes provistas por el usuario.

1.4 Resultados esperados

Los resultados del proyecto comprenden la obtención de un conjunto de imágenes de frutas reales y sintéticas de manzanas y mangos. El *dataset* será representativo de una variedad de defectos incluyendo podredumbre, costras, puntos negros y golpes, siendo estos algunos de los defectos encontrados con mayor frecuencia en manzanas y mangos. Las imágenes servirán como entrada para el entrenamiento y validación de los modelos de CNN para la detección de los defectos. Se espera optimizar modelos existentes para evaluar su rendimiento con métricas de desempeño y precisión en la detección de defectos.

1.5 Impacto comercial

El proyecto busca facilitar la verificación de la calidad externa de los productos durante las etapas posteriores a la cosecha como: selección, empaquetado y distribución. Se puede aumentar la capacidad de producción con modelos que realicen la clasificación a alta velocidad sin depender de criterios ambiguos. Además, los modelos de clasificación planteados son no invasivos o destructivos, y permitirían reducir los costos y tiempos de la inspección de calidad. Esta evaluación de modelos de *CNN* para detectar defectos en frutas puede ser usada en el desarrollo de herramientas tecnológicas para el control de calidad en el sector agrícola y de alimentos.

1.6 Marco Teórico

En esta sección se definirán los distintos tipos de defectos que se tienen en consideración en este estudio y sus causas correspondientes. Además, se mencionan las frutas objetivo y las investigaciones actuales sobre avances en cuanto a técnicas de detección de defectos aplicadas a estas frutas.

1.6.1 Defectos en frutas y hortalizas

Dentro de la industria alimentaria existen varios procesos de control de calidad de las frutas y hortalizas que determinan la calidad del alimento, basándose en ciertos parámetros tales como: color, olor, sabor, morfología y la presencia de defectos [2]. Cada

uno de estos aspectos es analizado en detalle para identificar puntos críticos en el aspecto de la fruta u hortaliza, y de esta manera determinar si es propensa a deteriorarse o perder su valor nutricional antes de llegar al consumidor final.

Para poder determinar qué tipo de defecto está asociado a una fruta u hortaliza, debemos agrupar las imperfecciones en categorías específicas que definan una posible causa en la aparición de defectos en los alimentos. En la Figura 1.1 se detallan las categorías que dividen a estos defectos, dependiendo de las prácticas de cosecha, la manipulación posterior y las condiciones de almacenamiento según [3]:

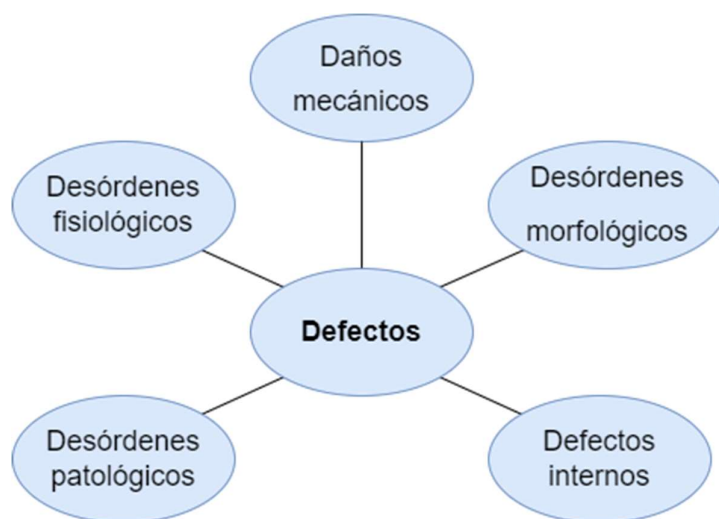


Figura 1.1 Tipos comunes de defectos en productos hortícolas (traducido al español) [3]

- **Desórdenes patológicos**

Son provocados por microorganismos y patógenos como virus, bacterias y hongos. La presencia de estos entes patógenos genera principalmente el aceleramiento de la maduración de la fruta y finalmente su putrefacción [3].

- **Daños mecánicos**

Los factores ambientales y la manipulación postcosecha son las principales causas de este tipo de daño. Se manifiesta como variaciones en la coloración externa del fruto y lesiones, estas afectaciones a su vez aceleran la maduración y facilita el desarrollo de infecciones [3].

- **Desórdenes fisiológicos**

Estos desórdenes se manifiestan con cambios en el sabor, textura y color del fruto y los causan la nutrición pobre de las plantas y temperaturas no adecuadas durante el desarrollo del fruto [3].

- **Desórdenes morfológicos**

Estos daños hacen referencia a la presencia de cambios en el aspecto normal y forma del fruto. Este tipo de defectos no suelen conllevar una alteración de las cualidades intrínsecas del fruto como su composición química, valor nutricional, olor, color, sin embargo, perjudican su estética [3].

- **Defectos internos**

Estos defectos son aquellos que no se manifiestan a simple vista y pueden ser precursores de otros tipos de desórdenes de tipo fisiológico, morfológico o daño mecánico [3].

1.6.2 Defectos en manzanas y mangos

Para efecto de este proyecto, se explorarán únicamente defectos en manzanas y mangos, los cuales se puedan analizar de manera visual y son los más comunes en las tareas de control de calidad para estas frutas.

Se han considerado los siguientes defectos:

- **Podredumbre**

Este defecto tiene origen en diversos factores como daños físicos provocados por el ambiente, insectos y también por infecciones de bacterias y hongos según [4]. Este defecto es observable a simple vista y provoca cambios en la coloración y textura de la piel de la fruta, la piel se torna color café, se oscurece progresivamente y pierde firmeza, como se observa en la Figura 1.2. Al aparecer en la etapa de postcosecha las infecciones que causan este defecto pueden seguir expandiéndose lentamente incluso durante el almacenamiento en condiciones ideales.



Figura 1.2 Manzana en estado de podredumbre [4]

- **Magulladuras y Golpes**

Según [5] los golpes en las frutas como el mango son producto de su manejo inadecuado durante la manipulación y empaquetado. En general el daño provocado por los golpes no atraviesa la piel, pero cambia la coloración y la textura de la parte afectada, como muestra la Figura 1.3. Este es un tipo de defecto considerado un daño mecánico.

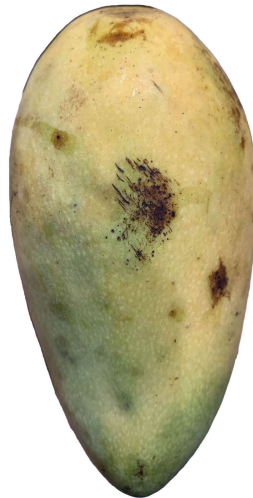


Figura 1.3 Mango con golpes [6]

- **Costras**

Las costras son una de las afecciones más comunes en las manzanas. Este tipo de defecto lo provocan infecciones por hongos tanto en las hojas como en la fruta, como indica [7]. A diferencia de un golpe, que no afecta las propiedades nutricionales y químicas de la fruta, las costras hacen al fruto no comestible. Este defecto se manifiesta

por medio de la piel resquebrajada, reseca y de color oscuro en el área afectada como se observa en la Figura 1.4.



Figura 1.4 Manzana con costras [7]

- **Puntos negros**

De acuerdo con [8] los puntos negros aparecen en el mango como síntoma de la infección por bacterias como *Xanthomonas axonopodis* pv. *mangiferaeindicae*. En la Figura 1.5 se pueden observar los síntomas de la infección que incluyen la aparición de lesiones en forma de estrella que con el pasar del tiempo se expanden y oscurecen. Incluso el mínimo grado de infección hace que la fruta pierda su calidad y potencial de venta.

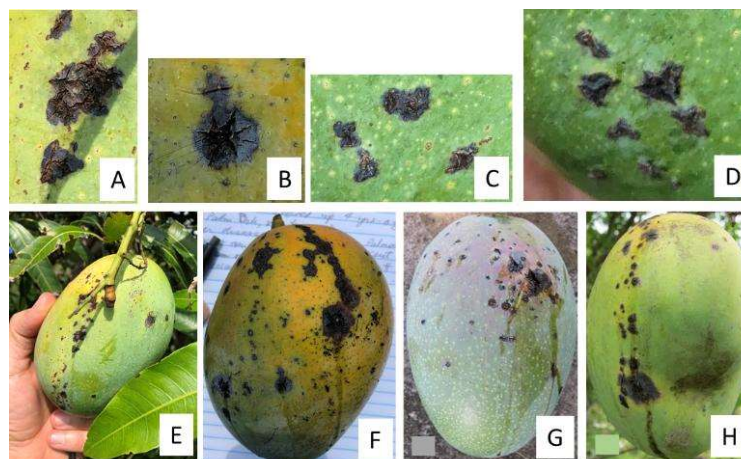


Figura 1.5 Puntos negros por infección bacteriana en mangos [8]

Para fines de esta investigación y el entrenamiento de los modelos *CNN* se considerarán imágenes de frutas con los siguientes defectos:

Tabla 1.1 Defectos considerados para la evaluación de modelos *CNN*

Fruta	Defectos
Manzana	Podredumbre
	Magulladuras/Golpes
	Costras
Mango	Podredumbre
	Magulladuras/Golpes
	Puntos negros

Los defectos seleccionados para el estudio como se muestra en la Tabla 1.1, fueron escogidos en base a la disponibilidad de datos para el entrenamiento de los modelos de *CNN* con bases de datos de acceso público y la revisión de otros estudios donde estos defectos también son objeto de análisis. Se determinan la podredumbre y golpes como defectos a analizar en común para ambas frutas. Además, se seleccionaron dos defectos que se presentan con más frecuencia en cada fruta, las costras en las manzanas y los puntos negros en los mangos.

1.6.3 Técnicas para la detección de defectos en frutas

La clasificación e inspección de la calidad de frutas y vegetales se realiza con el propósito de proveer al consumidor un producto sin defectos físicos externos perceptibles y con la garantía de que las propiedades químicas y el valor nutricional del producto no han sido alteradas. Las principales características externas evaluadas para asignar un grado de calidad a frutas y vegetales son el color, textura, tamaño, forma y la presencia de defectos. Estos indicadores son clave para la identificación de afecciones como enfermedades que pueden contaminar a producciones enteras como sugieren [9].

La investigación de [10] indica que el control de calidad de las frutas y vegetales se realiza comúnmente de forma manual, siendo personal capacitado quienes deben

identificar los defectos. Sin embargo, la intervención humana en este proceso lo vuelve susceptible a fallos ya sea por omisión, criterios ambiguos de clasificación y la cantidad de defectos que el personal debe ser capaz de reconocer.

[9] sostienen que los defectos como los golpes y podredumbre no siempre se manifiestan en un grado que permita diferenciarlos de una zona sana del fruto, lo que resulta un reto para las técnicas computacionales de evaluación que usan criterios como el color, textura y tamaño para realizar el análisis. El estudio de [11] propone un método de detección de defectos en frutas basado en *CNN*. En el estudio usaron un modelo propio y los modelos pre entrenados *AlexNet*, *Le-Net-5*, *VGG16* y *VGG19*, para detectar la podredumbre en manzanas, bananas y naranjas. Los resultados generados indican que este método es efectivo en la clasificación de frutas defectuosas y no defectuosas logrando obtener precisiones de 98.23% con su modelo propuesto y 90.81% *VGG16* el modelo con segundo mejor desempeño. En este estudio se encontró que los métodos de *transfer learning* fueron menos efectivos que el modelo propuesto, sin embargo, el entrenamiento de los modelos se realizó con solo 4035 imágenes de manzanas para las dos categorías “*fresh*” (1693) y “*rotten*” (2342).

1.6.4 Detección y clasificación de defectos en frutas por medio de visión por computador y modelos *CNN*

El uso de *CNN* es uno de los métodos más estudiados para la clasificación de frutas y la detección de defectos. [12] describen este tipo de redes neuronales que implementan capas de convolución donde se realizan operaciones matriciales entre la matriz de la imagen de entrada y una matriz de menor tamaño llamada *kernel*, para extraer las características de las imágenes de entrada. Estas redes también emplean capas de *pooling* con el propósito de reducir el tamaño de la salida de las capas de convolución. Algunos modelos de *CNN* aplican procesos de regularización para evitar el *overfitting*, por ejemplo, el *dropout*. Finalmente, las redes *CNN* realizan la clasificación por medio de una capa *fully connected* que recibe un vector de una dimensión o *flattened* como salida de las capas de convolución.

Según [12], las *CNN* se pueden usar para la detección de defectos en las frutas en las etapas previas y posteriores a la cosecha. En el caso de las manzanas una de las frutas sujeto de este estudio, usando el modelo *AlexNet* con 11 capas, se logró una precisión en la identificación de frutos defectuosos del 92.5% teniendo imágenes

hiperespectrales. En el estudio de [10], también se realizó detección de podredumbre en manzanas con imágenes *RGB* como entrada obteniendo el menor *MSE* y desviación estándar con el modelo *AlexNet*, seguido de *VGG11*.

Para la detección de defectos en mangos el modelo de red *MobileNetV2* logró una precisión promedio de 73.33% en la investigación de [13]. Los defectos detectados para la asignación de un grado de calidad fueron la presencia de puntos negros y manchas en el exterior de la fruta.

En las manzanas los golpes se pueden evidenciar por medio de manchas en el exterior de la fruta y en el estudio de [14] el modelo *DenseNet121* resulta el más efectivo con un *MCC* de 0.978% realizando el análisis en un dataset de imágenes *RGB* y realizando la comparación con los modelos *AlexNet*, *ResNet-18*, *ResNet-50* y *VGG19*.

En los experimentos de [15] se obtuvo uno de los resultados más prometedores para identificar defectos relacionados a la podredumbre en tanto manzanas y mangos, alcanzando la exactitud de 97.34%, *recall* de 0.97, precisión de 0.97 y *F1-score* de 0.97, siendo estas las métricas más altas alcanzadas con el modelo *InceptionV3*.

A continuación, en la Tabla 1.2 se detallan los resultados de la revisión del estado del arte para la detección de defectos en frutas usando CNN, registrando los modelos, métricas y datos usados en estos estudios.

Tabla 1.2 Comparación de modelos de CNN en la detección de defectos en manzanas y mangos

Fruta	Defectos	Modelos	Desempeño
Manzana [16]	Podredumbre	<i>ResNet-50</i>	97.00% - Accuracy
Manzana Mango [15]	Podredumbre	<i>InceptionV3</i> <i>Xception</i> <i>VGG16</i> <i>MobileNet</i> <i>NASNetMobile</i>	97.34% - Accuracy 97.16% - Accuracy 96.54% - Accuracy 95.57% - Accuracy 75.29% - Accuracy
Manzana [11]	Podredumbre	<i>VGG16</i> <i>VGG19</i> <i>LeNet-5</i> <i>AlexNet</i>	90.81% - Accuracy 76.48% - Accuracy 82.93% - Accuracy 83.56% - Accuracy
Manzana [10]	Podredumbre	<i>GoogleNet</i> <i>VGG11</i> <i>AlexNet</i> <i>ResNet</i>	4.404 - MSE 3.934 - MSE 4.099 - MSE 4.058 - MSE
Manzana [14]	Golpes Manchas	<i>VGG19</i> <i>RestNet-50</i> <i>DenseNet121</i>	0.969 - MCC 0.970 - MCC 0.997 - MCC
Mango [13]	Puntos negros	<i>SqueezeNet</i> <i>ResNet-50</i> <i>Enhanced</i> <i>MobileNetV2</i>	96.94% - Accuracy 86.67% - Accuracy 96.67% - Accuracy 73.33% - Accuracy
Manzana [17]	Costras	<i>VGG16</i>	87.50% - Accuracy

Los estudios presentados difieren en el tamaño de sus bases de datos, la división de los datos para entrenamiento y validación, los hiper-parámetros seleccionados y los defectos que abarcan en sus imágenes de entrada. Estos estudios limitan las bases de datos a menos 4000 imágenes en promedio y menos de 2000 imágenes por categoría. Por otro lado, los defectos abarcados en los estudios mencionados se limitan a categorizar las imágenes de entrada en de acuerdo con la presencia o no de defectos, lo que puede influenciar en la capacidad de los modelos de detectar ciertos tipos de defectos con menor presencia en el *dataset*.

Según la literatura consultada también se ha realizado una selección de modelos de CNN para realizar la evaluación de su desempeño identificando defectos en manzanas y mangos.

Tabla 1.3 Modelos CNN escogidos por su desempeño en detección de defectos

Fruta	Defecto	Modelo CNN
Manzana	Podredumbre	<i>InceptionV3</i>
	Golpes/Magulladuras	<i>DenseNet121</i>
	Costras	<i>VGG16</i>
Mango	Podredumbre	<i>InceptionV3</i>
	Golpes/Magulladuras	<i>VGG16</i>
	Puntos negros	<i>MobileNetV2</i>

Otra limitante para la evaluación de los modelos encontrada en la revisión del estado del arte son las métricas bajo las que se comparan los modelos. Principalmente se usa sólo el accuracy para evaluar los modelos e incluso es la única métrica presentada. Por esta razón el presente estudio evaluará los modelos bajo las métricas de *accuracy*, *precision*, *recall*, *F1-score*, matriz de confusión.

1.6.5 Técnicas de adquisición de imágenes

Para la adquisición de imágenes reales y sintéticas de frutas con el fin de identificar defectos existen varios métodos de adquisición según las características de la fruta que se desea analizar.

Para el caso de imágenes reales, para explorar defectos superficiales que se manifiestan en cambios de color y textura de la piel de la fruta se usan cámaras en el dominio *RGB*, como en el estudio de [18]. También es posible usar cámaras hiperespectrales como lo realizaron [19] donde obtuvieron imágenes para detectar costras, golpes, podredumbre entre otros defectos en manzanas.

En el caso de las imágenes sintéticas, se pueden destacar algunas técnicas para crear este tipo de imágenes. La generación de *datasets* sintéticos no es una práctica nueva, en [20] se usó 3000 imágenes generadas en un ambiente virtual con el *software* *CARLA* para la estimación de poses de cámara usando la arquitectura de CNN ResNet-50 con modificaciones. También, por medio del *Ray Tracing* se puede recrear objetos en

ambientes virtuales usando las propiedades de los rayos de luz. La propuesta de [21] consiste en modelar objetos en 3D para usar software que permita crear una visualización realista del objeto luego de realizar cálculos en base al comportamiento de la luz dirigida hacia él desde distintos ángulos y fuentes. Esta técnica persigue este fotorrealismo utilizando leyes naturales de la propagación de luz, reflexión y refracción, que dependen de los materiales.

[22] indica que otros métodos para generar imágenes sintéticas incluyen el modelado y renderizado de los objetos en herramientas como *Blender* y *Unreal Engine*, permitiendo un mayor control sobre las características y condiciones deseadas en el *dataset*. Dado que el modelado 3D desde cero puede ser un proceso lento y costoso, una alternativa para generar datos sintéticos es la captura de los modelos 3D usando sensores *LiDAR* y aplicaciones que usan los datos de estos sensores para reconstruir los objetos en un ambiente virtual.

Por último, existe la posibilidad de utilizar herramientas recientemente publicadas para generar imágenes sintéticas usando sistemas de aprendizaje de máquina como son los modelos *DALL-E* y *DALL-E mini*. Este tipo de sistemas son capaces de generar imágenes en alta resolución (en el caso de *DALL-E*) a partir de texto de entrada que describen las condiciones deseadas y los detalles de los objetos o escenas, según [23].

En este trabajo de investigación la generación del dataset se realizó recopilando imágenes reales obtenidas con sensores *RGB* y disponibles en bases de datos públicas y adicionalmente, se generaron imágenes sintéticas usando *DALL-E mini* a partir de descripciones cortas de las frutas con los defectos seleccionados para el estudio.

En el siguiente capítulo, se detalla la metodología utilizada para la recolección de los datos, el entrenamiento y la evaluación de los modelos. Se presentarán las bases de datos seleccionadas y el proceso de refinamiento, aumento de datos. Asimismo, se detalla la generación de la base de datos de imágenes sintéticas. Además, se presentará la arquitectura de los modelos seleccionados y el proceso de entrenamiento con los hiper-parámetros ajustados.

CAPÍTULO 2

2. METODOLOGÍA

En el siguiente capítulo se describe la metodología utilizada para la obtención del *dataset* y el entrenamiento de los modelos *CNN* seleccionados para la identificación de frutas defectuosas. Se detallan aspectos sobre el proceso realizado para llegar a la cantidad meta en imágenes reales y sintéticas de manzanas y mangos, ambas frutas objetivo del presente estudio. Adicionalmente, se muestra una breve descripción sobre cada uno de los modelos escogidos para el entrenamiento y posterior evaluación.

2.1 Módulos de desarrollo de la solución

Se decidió dividir la metodología en tres módulos, como se observa en la Figura 2.1. El módulo 1, consiste en la generación del *dataset* donde se recolectarán imágenes reales de los defectos y se generarán imágenes sintéticas para complementar el *dataset*, además se va a refinar y aumentar la cantidad de imágenes con técnicas de *data augmentation*. El segundo módulo se dedicó a la realización de los entrenamientos de los modelos *CNN* seleccionados para obtener las métricas de rendimiento utilizadas en el tercer módulo que es de evaluación de los modelos. Finalmente, en el cuarto módulo se desarrolló un prototipo de aplicación *web* para obtener resultados sobre los defectos en imágenes de frutas con y sin defectos provistas por el usuario.

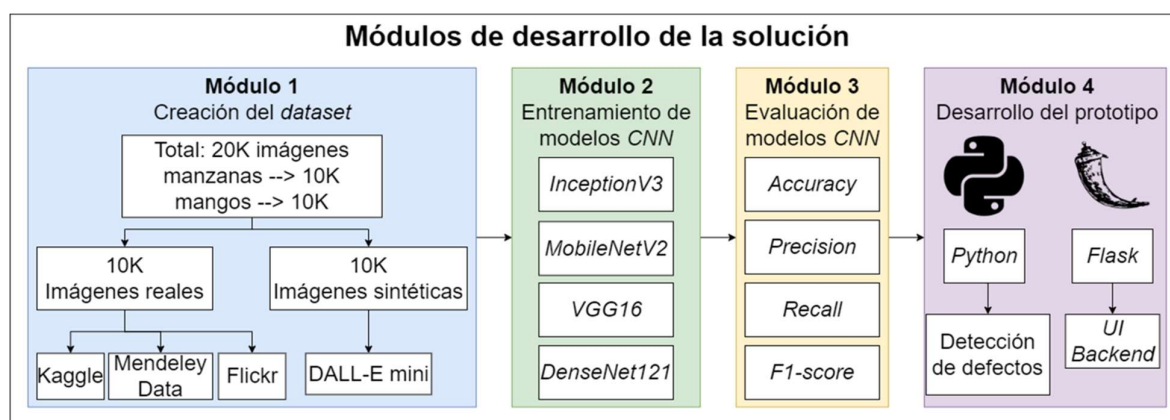


Figura 2.1 Módulos de desarrollo de la solución

2.2 Módulo 1: Creación del dataset

En esta sección se detalla la metodología usada para la recopilación de imágenes para los *datasets* de imágenes reales y sintéticas, incluyendo las técnicas de generación de imágenes sintéticas y el posterior refinamiento del dataset.

2.2.1 Recopilación de imágenes reales y generación de imágenes sintéticas

En esta etapa, con base en la literatura, se diseñó un diagrama organizacional de directorios, se indica la cantidad y a qué grupo pertenece cada sección de imágenes, como se presenta en la Figura 2.2.

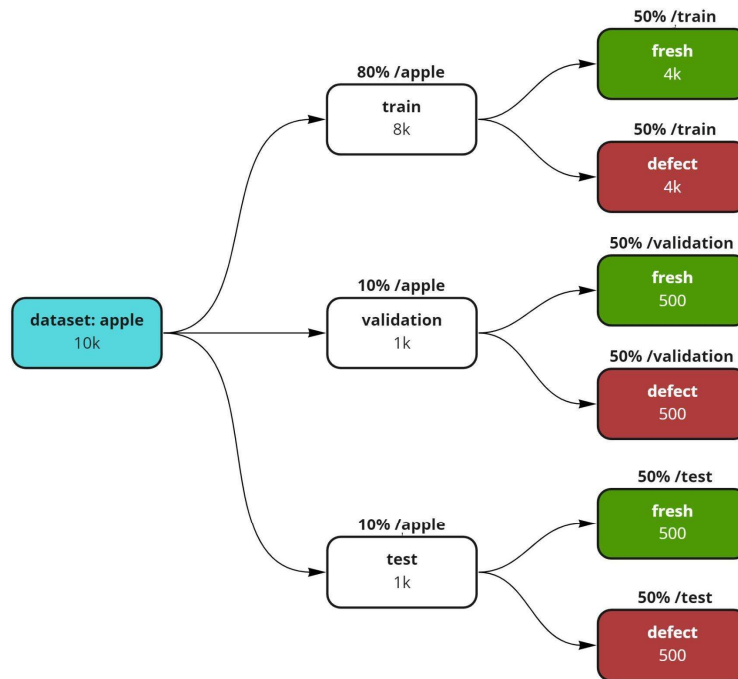


Figura 2.2 Estructura del *dataset*

Como se muestra en la Figura 2.2 los directorios se organizaron por fruta (manzana y mango), por tanto, el directorio del dataset de manzanas contó con 10000 imágenes en total y la misma cantidad para el directorio de mangos. La partición usada para los entrenamientos y validación es del 80% para entrenamiento (*training*), 10% validación (*validation*) y 10% pruebas (*testing*).

Para llevar a cabo la evaluación del desempeño de los modelos de *CNN* seleccionados se definió una estrategia para recopilar las imágenes de entrada de las

frutas. El *dataset* recolectado consta de imágenes de frutas reales y de imágenes sintéticas. Las imágenes reales fueron recopiladas de diversas bases de datos públicas como *Kaggle*, *Flickr* y *Mendeley Data*.

En la Tabla 2.1 se resumen las bases de datos consultadas y utilizadas para generar el *dataset* de imágenes reales.

Tabla 2.1 Fuentes de imágenes reales para creación del dataset

Base de Datos	Fruta	Categoría
<i>Fruits fresh and rotten for classification</i> [24]	Manzana	Sin defectos Podredumbre
<i>Fruits-262</i> [25]	Manzana Mango	Sin defectos
<i>Mango Varieties Classification and Grading</i> [26]	Mango	Sin defectos Puntos negros
<i>Sala Mango Images Dataset</i> [6]	Mango	Sin defectos Golpes
<i>AppleScabFDs</i> [27]	Manzana	Costras
<i>Spoiled and fresh fruit inspection dataset</i> [28]	Mango	Podredumbre
<i>Mango Dataset - Studio Setup</i> [29]	Mango	Sin defectos Podredumbre
<i>Kesar Mango</i> [30]	Mango	Podredumbre
<i>Mango: Bacterial black spot</i> [31]	Mango	Puntos negros
<i>Apple fruit disease</i> [32]	Manzana	Sin defectos Costras

El motivo para la generación de imágenes sintéticas fue aumentar el volumen de imágenes de entrada dada la baja disponibilidad de *datasets* públicos con imágenes específicas de los defectos seleccionados para la clasificación. La generación del *dataset* de imágenes sintéticas se realizó usando el modelo *text-to-image* DALL-E mini. Este modelo basado en el modelo de codificación de lenguaje *BART* y el modelo de decodificación de lenguaje *VQGAN* puede generar imágenes a partir de un texto corto, de acuerdo a [33]. Usando la aplicación web disponible en la comunidad *HuggingFace* fue posible obtener imágenes sintéticas para las categorías de defectos de ambas frutas. En la Tabla 2.1 se detallan las entradas de texto utilizadas para generar el *dataset* sintético.

Tabla 2.2 Entradas de texto para la generación de imágenes con *DALL-E mini*

Fruta	Clase	Entradas de texto
Manzana	Sin defectos	<i>"apple", "red apple", "golden apple", "healthy apple", "nice apple"</i>
	Podredumbre	<i>"rotten apple"</i>
	Golpes	<i>"bruised apple", "bruised red apple", "bruised golden apple"</i>
	Costras	<i>"apple scabs", "red apple scabs", "golden apple scabs", "apple fungus", "apple anthracnose", "scab apple"</i>
Mango	Sin defectos	<i>"mango", "red mango", "green mango"</i>
	Podredumbre	<i>"rotten mango"</i>
	Golpes	<i>"bruised mango"</i>
	Puntos negros	<i>"mango black spots", "mango bacterial black spots"</i>

Si bien existen varias formas de obtener imágenes del defecto que se está estudiando, es importante notar que las imágenes tengan variedad tanto en orientación, presentación de síntomas, distancia, entre otros. Escogiendo como ejemplo el defecto de podredumbre en las manzanas, en la Figura 2.3, se puede evidenciar un ejemplo de las imágenes generadas con *DALL-E mini*.



Figura 2.3 Imágenes sintéticas generadas con *DALL-E mini* para el defecto de podredumbre en manzanas

2.1.2 Refinamiento y preprocesamiento de imágenes

Dada la variedad en calidad de los *datasets* obtenidos y generados fue necesario refinarlos. Los *datasets* de imágenes reales fueron explorados manualmente para eliminar imágenes que no correspondan a las clases de defectos seleccionados e imágenes de baja calidad. Además, en ciertos *datasets* fue necesario editar el tamaño de las imágenes para que solo contengan la región de interés, es decir, las frutas.

Por la falta de *datasets* de imágenes reales para los defectos específicos fue necesario aplicar técnicas de *data augmentation* como la aplicación aleatoria de transformaciones, cambios de contraste y rotaciones a las imágenes usando las librerías *OpenCV* y *Albumentations* por medio de scripts de *Python*.

En el caso del *dataset* sintético el refinamiento se realizó por medio de la selección manual de las imágenes generadas por *DALL-E mini* debido a la variedad de imágenes generadas en cada lote y su relevancia del texto descriptivo dado. Finalmente, se estandarizó el tamaño de las imágenes a 256x256 píxeles para reducir el tamaño del *dataset*.

Aunque la clasificación que se realizó durante el experimento es binaria, es decir, se clasificaron las imágenes de acuerdo con la presencia o ausencia de defectos en las frutas, se balancearon las cantidades de imágenes de entrada por cada tipo de defecto

seleccionado. Se buscó generar un *dataset* donde los defectos seleccionados se representan de forma equitativa para mejorar la capacidad de los modelos de detectar frutas con defectos diferentes a la podredumbre, como se muestra en la Tabla 2.3.

Tabla 2.3 Distribución del dataset

Fruta	Clase	Tamaño de <i>dataset</i>			
		Imágenes reales	Imágenes sintéticas	Total por categoría	Total
Manzana	Sin defectos	2500	2500	5000	10000
	Podredumbre	1000	1000	2000	
	Golpes	750	750	1500	
	Costras	750	750	1500	
Mango	Sin defectos	2500	2500	5000	10000
	Podredumbre	1000	1000	2000	
	Golpes	750	750	1500	
	Puntos negros	750	750	1500	

Para el entrenamiento se usará una partición de 80/10/10 del *dataset* para entrenamiento, validación y pruebas respectivamente.

2.3 Módulo 2: Entrenamiento de modelos de CNN

Las pruebas de los modelos y la evaluación respectiva se llevaron a cabo en el entorno web de *Kaggle* donde de forma gratuita se puede hacer uso de máquinas virtuales con *GPUs* para acelerar el entrenamiento de los modelos. En este entorno se hace uso de *Jupyter Notebooks* para realizar los scripts con el lenguaje *Python* y librerías como *Tensor Flow* para cargar el *dataset*, entrenar los modelos y obtener métricas de rendimiento. Durante los entrenamientos se aplicó *transfer learning*, para cargar los pesos de los modelos seleccionados y aplicar sus arquitecturas al problema de este

estudio que es de clasificación binaria. Los modelos seleccionados para este estudio fueron *MobileNetV2*, *InceptionV3*, *VGG16* y *DenseNet121*.

Por medio de las librerías *Keras* y *Tensor Flow* es posible cargar los modelos y aplicarle modificaciones. Primero, se cargaron los modelos usando los pesos que obtuvieron al ser entrenados con el *dataset ImageNet*. Las capas de salida fully connected fueron removidas, dado que se usaban para clasificación multiclase y era necesario ajustarlas al problema actual de clasificación binaria.

A continuación, se detallan las arquitecturas de los modelos entrenados:

2.3.1 MobileNetV2

El principal objetivo de este modelo de red convolucional es el de obtener el mejor rendimiento en dispositivos móviles, basada en una estructura residual invertida, donde las conexiones residuales se encuentran en las capas con pocos nodos. Su capa intermedia de expansión utiliza convoluciones *depth wise* ligeras, para filtrar características como una fuente de no-linealidad [34].

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Figura 2.4 Arquitectura de *MobileNetV2* [34]

Como se observa en la Figura 2.4, su arquitectura está compuesta por una red convolucional *full connected* de 32 filtros, seguido por 19 capas residuales de pocos nodos. Según [34], lo característico de este modelo, es que tiene un acercamiento estrecho-amplio-estrecho, es decir, al inicio expande la red usando filtros 1x1 en la capa de convolución, dado que las capas siguientes con filtros 3x3, ya realizan la reducción de los números de parámetros para concordar con la cantidad inicial del número de canales. La idea general es que esta red se comporta como un bloque residual invertido,

ya que existen conexiones *skip* entre las partes más estrechas de la red, que es lo contrario a como una red con conexiones residuales se comporta.

2.3.2 DenseNet121

La arquitectura *DenseNet121* hace uso de conexiones densas para lograr generar menos parámetros sin sacrificar el *accuracy*. Esta arquitectura usa los llamados bloques densos, donde cada capa es capaz de recibir información de todas las capas anteriores. Haciendo uso de la concatenación se puede obtener un “conocimiento colectivo” como indica [35]. Estos cambios respecto a otros modelos le permiten reducir el uso de memoria y ser más eficiente. En la Figura 2.5 se puede observar la arquitectura de la red DenseNet121 con sus bloques densos.

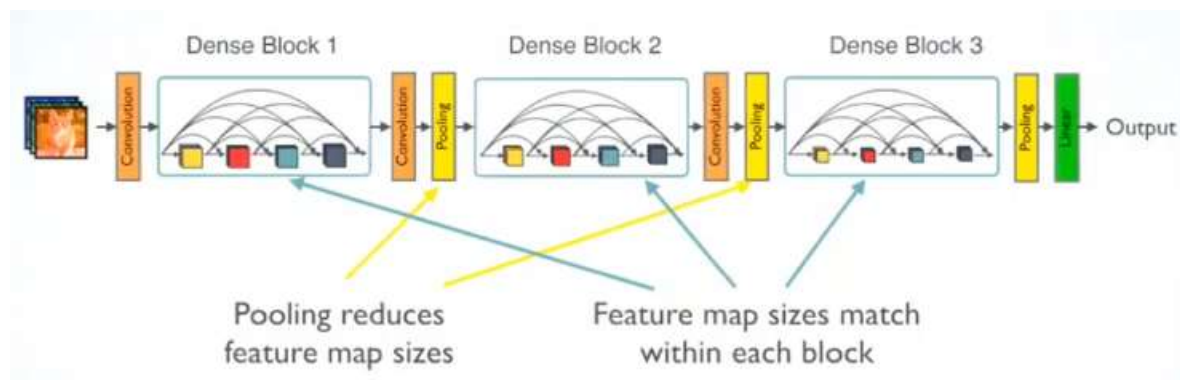


Figura 2.5 Arquitectura de DenseNet121 [35]

DenseNet121 combina los bloques densos con capas de convolución y capas de *bottleneck*, que son convoluciones con *kernel* más pequeño para reducir el número de parámetros. Uno de los inconvenientes más frecuentes en la construcción de una arquitectura de red profunda, es que, a medida que aumenta el número de capas *CNN*, surge el problema del “gradiente de fuga”, es decir, que a medida de que aumenta el recorrido de la información desde las capas de entrada hasta las de salida, hay probabilidades de perder información o que esta se “desvanezca”, lo que puede reducir la efectividad del entrenamiento. Con esto en mente, las *DenseNets* proveen una arquitectura donde se simplifica el patrón de conectividad entre las capas. La principal ventaja de usar los bloques densos es que al obtener una especie de historial de *features*

es posible que estos sean más diversos y se evita el problema del gradiente desvaneciente, como indica [35].

2.3.3 InceptionV3

La arquitectura *Inception* propuso la idea de múltiples filtros de diferentes tamaños en el mismo nivel para evitar el *overfitting*. El modelo se compone de capas paralelas en lugar de capas profundas. Los modelos iniciales contaban con cuatro capas paralelas: 3 capas de convolución, 1x1, 3x3 y 5x5, una capa de *max pooling* de 3x3. El modelo actual *InceptionV3* cuenta con 42 capas donde se aplican técnicas como la factorización en convoluciones más pequeñas y las convoluciones asimétricas para reducir el tamaño de los filtros sin sacrificar la efectividad y rendimiento del modelo de acuerdo a [36].

En las Figuras 2.6 y 2.7 se detalla el cambio en la arquitectura de las capas paralelas en el modelo *InceptionV3* y cómo se ha reducido el tamaño de los filtros usados.

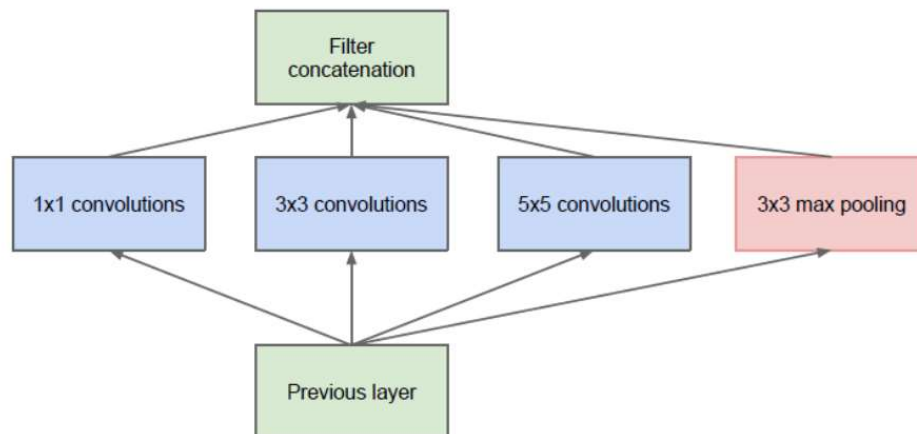


Figura 2.6 Capas paralelas en el modelo *Inception* original [36]

El primer modelo de *Inception* usaba varios filtros de convolución en la misma capa como se muestra en la Figura 2.5, sin embargo, los filtros de 5x5 requerían demasiados recursos computacionales.

Según [36], las versiones posteriores de *Inception* mejoraron la eficiencia reduciendo el tamaño de los filtros y usando filtros asimétricos que lograban la misma cobertura en la extracción de características, pero eran menos costosos

computacionalmente. En la Figura 2.7 se evidencia que incluso los filtros 3x3 se descomponen en filtros más pequeños.

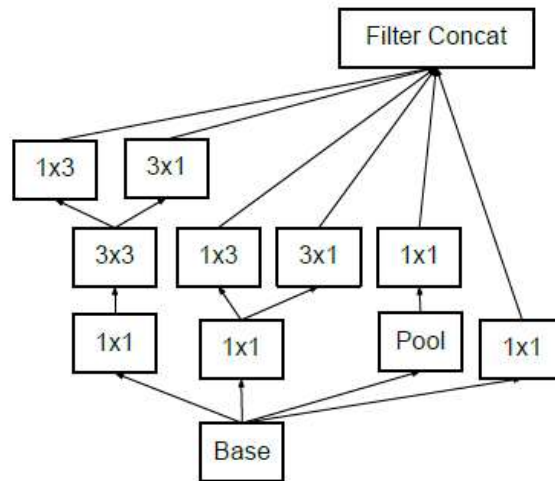


Figura 2.7 Capas paralelas en el modelo *InceptionV3* [36]

2.3.4 VGG16

VGG16 es un modelo de arquitectura *VGG* con 16 capas. Este modelo se caracteriza por la uniformidad de sus capas de convolución y más pooling dado que todas las capas de convolución usan filtros 3x3 y las capas de max pooling usan filtros de 2x2. Este modelo cuenta con aproximadamente 138 millones de parámetros y fue entrenado usando *ImageNet*, una base de datos con millones de imágenes. *VGG16* presenta varias mejoras respecto a *AlexNet* sobre todo en los tamaños de filtros que a su vez mejoran la velocidad de entrenamiento. Otra mejora desarrollada fue el uso de las unidades de activación *ReLU* que reducen el tiempo de entrenamiento [37].

En la Figura 2.8 se describe la composición de capas del modelo *VGG16* con sus capas de convolución, *pooling* y clasificación.

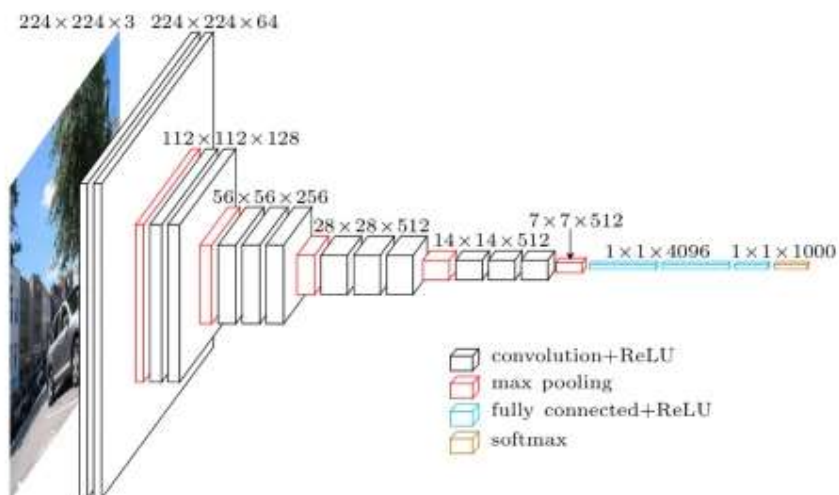


Figura. 2.8 Capas del modelo VGG16 [38]

El número 16 hace referencia solo a las capas con pesos, es decir las de convolución y las capas densas. Las capas de convolución y *pooling* se encuentran distribuidas consistentemente en toda la arquitectura como menciona [37].

Para cada entrenamiento se capturó información destinada al desarrollo de los siguientes módulos. Cada entrenamiento dio como resultado un modelo en archivo *.h5* que sirve para exportar e importar el modelo, este archivo se generó a partir del punto en el entrenamiento donde se obtuvo el pico de rendimiento en *accuracy*. También se capturaron las fluctuaciones de las métricas a lo largo de las épocas y se guardaron en archivos CSV.

2.3.5 Selección de hiper-parámetros

A partir de la revisión del estado del arte, que comprende el uso diferentes arquitecturas para la detección de defectos en frutas, se extrajo la siguiente tabla con los hiper-parámetros presentes en mayor frecuencia en el entrenamiento de los modelos utilizando las arquitecturas mencionadas anteriormente.

Tabla 2.4 Hiper-parámetros de entrenamiento

Optimizador	Épocas	Lotes	Taza de aprendizaje
RMSprop	10, 20, 30	16	0.001

Las modificaciones por considerar dependerán del avance en el entrenamiento y la evaluación de la tasa de error realizando el reajuste interno de los parámetros. El número de capas de convolución, transición y precisión, serán determinadas inicialmente por la misma arquitectura del modelo pre entrenado.

Para la tarea de clasificación binaria es necesario ajustar las capas *fully connected* de salida de los modelos. Para todos los modelos se usó una capa de densa con 1024 unidades y función de activación *ReLU*, seguida de una capa de dropout, luego otra capa densa con 512 unidades y función de activación *ReLU*. Finalmente, se usa una capa de 2 unidades y función de activación *softmax* para obtener el vector *one hot encoded* con la clasificación.

Se decidió estudiar el efecto de la capa de dropout en las métricas de rendimiento obtenidas por los modelos, por lo cual se realizaron tres tipos de ensayos, uno sin usar la capa de dropout y los otros dos con *dropout rate* de 0.2 y 0.5, con el fin de mitigar el *overfitting*.

2.4 Módulo 3: Evaluación de modelos de CNN

Las métricas, reflejadas con su respectiva ecuación en la Figura 2.9, se usaron para medir los resultados de cada entrenamiento, serán la precisión, el recall (exhaustividad), la exactitud (*accuracy*), el *F1-score*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Figura 2.9 Ecuaciones de las métricas [39]

- La precisión define el porcentaje de frutas defectuosas clasificadas que en realidad son defectuosas y también define la calidad de los resultados.
- El *recall* representa el porcentaje de frutas defectuosas que se pueden identificar del total de frutas defectuosas.
- La exactitud muestra el porcentaje de frutas defectuosas y no defectuosas identificadas correctamente.
- El *F1-score* es la relación entre el *recall* y la precisión

2.5 Módulo 4: Desarrollo del prototipo

El prototipo que se desarrolló está pensado para su uso en la *web*. La arquitectura para este prototipo se basa en *Python* que permite interactuar con los modelos resultantes de los entrenamientos y el uso de las librerías para aprendizaje de máquina. Por otro lado, se usó el *framework Flask* con el que se desarrolló la interfaz gráfica en formato de formulario donde el usuario puede seleccionar la imagen de entrada y el modelo con el que quiere realizar la detección de defectos. El prototipo devuelve al usuario la imagen analizada junto con el resultado sobre la presencia o ausencia de defectos.

En el siguiente capítulo se ilustran los resultados obtenidos al finalizar cada entrenamiento de los modelos *MobileNetV2*, *InceptionV3*, *VGG16* y *DenseNet121*, recibiendo como entradas las imágenes de mangos y manzanas, cada fruta clasificada como sin defecto o con defecto, dando como resultado 9 entrenamientos para cada modelo.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En este capítulo se detallarán los resultados obtenidos luego de la evaluación del rendimiento de los modelos seleccionados en base a los entrenamientos realizados con la metodología descrita en el capítulo anterior.

Para cada fruta se realizó un total de 36 entrenamientos con variaciones en los modelos *CNN*, el número de épocas y *dropout rates* utilizados. A partir de cada entrenamiento se obtuvieron las métricas de validación: *Accuracy*, *Precision*, *Recall* y *F1-score*, así como una métrica adicional de *Accuracy* con la porción del *dataset* destinada a pruebas (*testing*).

Tanto para los entrenamientos de los modelos evaluados con mangos como para manzanas se aplicaron los mismos hiper-parámetros, optimizador *RMSprop* con *learning rate* de 0.001 además el tamaño de entrada de las imágenes se ajustó a 256x256 píxeles. Se emplearon dos capas densas, la primera con 1024 unidades y la segunda con 512 unidades, ambas con función de activación *ReLU*. Finalmente, se hizo uso de una capa con 2 unidades y función de activación *softmax* para obtener las salidas binarias.

3.1 Resultados con el dataset de imágenes de manzanas

En las Tablas 3.1, 3.2, 3.3 y 3.4 se detallan los resultados obtenidos para el *dataset* de imágenes de manzanas con y sin defectos para los cuatro modelos seleccionados y las variaciones de *dropout rate* y épocas. Se resalta la fila correspondiente al mejor resultado obtenido en el *test accuracy* para cada modelo.

Tabla 3.1 Métricas de rendimiento de *InceptionV3* para el dataset de imágenes de manzanas

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.7995	0.4483	0.9820	0.9820	0.9820	0.7620
10	0.2	0.8315	0.4040	0.9810	0.9810	0.9810	0.7830
10	0.5	0.5175	0.3916	0.9600	0.9600	0.9600	0.4765
20	-	0.8825	0.3649	0.9840	0.9840	0.9840	0.8815
20	0.2	0.9085	0.4354	0.9840	0.9840	0.9840	0.9205
20	0.5	0.7125	0.3242	0.9800	0.9800	0.9800	0.6510
30	-	0.8850	0.4715	0.9860	0.9860	0.9860	0.8950
30	0.2	0.9240	0.8458	0.9810	0.9810	0.9810	0.9345
30	0.5	0.8405	0.6971	0.9730	0.9730	0.9730	0.8455

Tabla 3.2 Métricas de rendimiento de *MobileNetV2* para el dataset de imágenes de manzanas

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.9605	0.4519	0.9890	0.9890	0.9890	0.9425
10	0.2	0.9400	0.3407	0.9890	0.9890	0.9890	0.9455
10	0.5	0.9085	0.2578	0.9840	0.9840	0.9840	0.9000
20	-	0.9560	0.4370	0.9890	0.9890	0.9890	0.9560
20	0.2	0.9710	0.6131	0.9860	0.9860	0.9860	0.9760
20	0.5	0.9440	0.2561	0.9890	0.9890	0.9890	0.9535
30	-	0.9275	0.2692	0.9870	0.9870	0.9870	0.9315
30	0.2	0.9695	0.4285	0.9870	0.9870	0.9870	0.9790
30	0.5	0.9690	0.3884	0.9870	0.9870	0.9870	0.9740

Tabla 3.3 Métricas de rendimiento de VGG16 para el dataset de imágenes de manzanas

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.4655	0.1592	0.9700	0.9700	0.9700	0.4645
10	0.2	0.5450	0.1213	0.9840	0.9840	0.9840	0.5180
10	0.5	0.2430	0.0861	0.9720	0.9720	0.9720	0.1915
20	-	0.6745	0.1092	0.9880	0.9880	0.9880	0.6520
20	0.2	0.6225	0.1995	0.9840	0.9840	0.9840	0.6120
20	0.5	0.4575	0.0810	0.9850	0.9850	0.9850	0.3925
30	-	0.7665	0.1903	0.9850	0.9850	0.9850	0.7645
30	0.2	0.8105	0.8780	0.9630	0.9630	0.9630	0.8175
30	0.5	0.6565	0.1069	0.9840	0.9840	0.9840	0.6675

Tabla 3.4 Métricas de rendimiento de DenseNet121 para el dataset de imágenes de manzanas

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.8605	0.1130	0.9890	0.9890	0.9890	0.8505
10	0.2	0.8945	0.1646	0.9910	0.9910	0.9910	0.8400
10	0.5	0.6400	0.1318	0.9800	0.9800	0.9800	0.4120
20	-	0.9085	0.1599	0.9910	0.9910	0.9910	0.9140
20	0.2	0.9285	0.1134	0.9900	0.9900	0.9900	0.9310
20	0.5	0.7245	0.0601	0.9880	0.9880	0.9880	0.6280
30	-	0.9330	0.0585	0.9940	0.9940	0.9940	0.9285
30	0.2	0.9385	0.1423	0.9940	0.9940	0.9940	0.9320
30	0.5	0.8880	0.1283	0.9900	0.9900	0.9900	0.8775

En los resultados obtenidos para estos entrenamientos se puede encontrar una tendencia en cuanto al *accuracy* y la combinación de épocas y *dropout rate*. Las mejores métricas para el *accuracy* del *dataset* de pruebas se lograron con los entrenamientos de 30 épocas y *dropout rate* de 0.2.

En este punto es importante recordar que, para efectos del estudio, se contemplan los 3 defectos que tuvieron mayor incidencia durante el proceso de creación del *dataset*. Para el caso de la manzana, se consideraron todas aquellas que presentaran golpes, podredumbre y costras, en gran medida o incidencias apenas visibles. En la Figura 3.1 se puede observar una muestra de cada defecto que se consideró para el entrenamiento.

Los resultados del entrenamiento, tanto en la fase de validación como de prueba, fueron obtenidos a partir del procesamiento de estos 3 tipos de defectos, los cuales enriquecen la variedad de imperfecciones que se analizaron en este estudio. Para efectos de este estudio, al buscar determinar si una manzana presentaba defectos, no se realizó una categorización por cada defecto, en su lugar, se buscó analizar la capacidad de detección de defectos que poseían todos los modelos escogidos, al identificar la presencia o no de un defecto de cualquier tipo.

En la Figura 3.1 se evidencia la tendencia de que al aumentar el número de épocas aumenta el *accuracy*. De la misma manera aumentar el dropout a 0.2 en comparación con no aplicar esta técnica de normalización permitió que los modelos mejoren su rendimiento. Aplicar un dropout rate superior a 0.5 provoca que el *accuracy* disminuya, posiblemente por la pérdida de información en las capas densas.

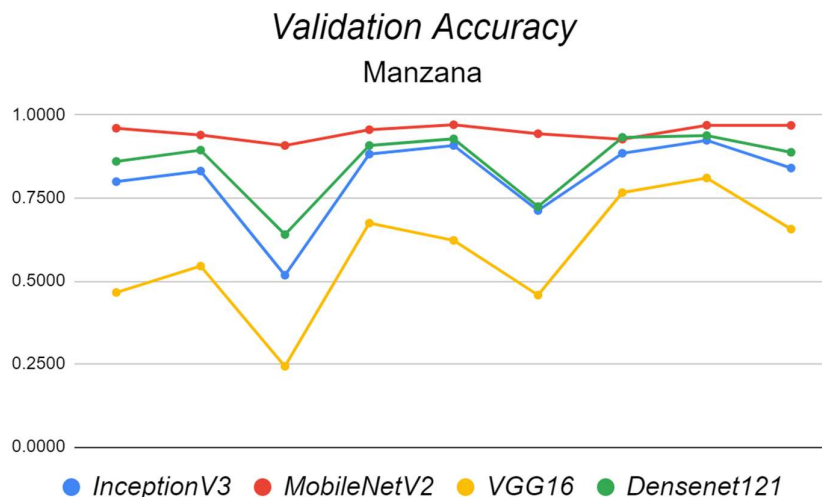


Figura 3.1 *Validation accuracy* para los entrenamientos con el dataset de imágenes de manzana

El modelo CNN *MobileNetV2* registró el *accuracy* más alto con el dataset de prueba con un 97.90%. Los modelos *InceptionV3* y *DenseNet121* presentaron porcentajes similares en esta métrica con 93.40% y 93.20% respectivamente. El modelo *VGG16* presentó el *accuracy* más bajo con 81.75%.

Otras métricas que comprueban la efectividad de los modelos son la precisión y el *recall*, con las cuales se obtuvo valores cercanos a 1 desde épocas tempranas, indicando que los modelos son capaces de realizar la clasificación sin falsos positivos o falsos negativos. Consecuentemente, la métrica de *F1-score* obtenida para los modelos también fue alta dado que representa la relación entre la precisión y el *recall*. En las siguientes gráficas se evidencia la tendencia de los modelos entrenados en cuanto a las métricas mencionadas anteriormente.

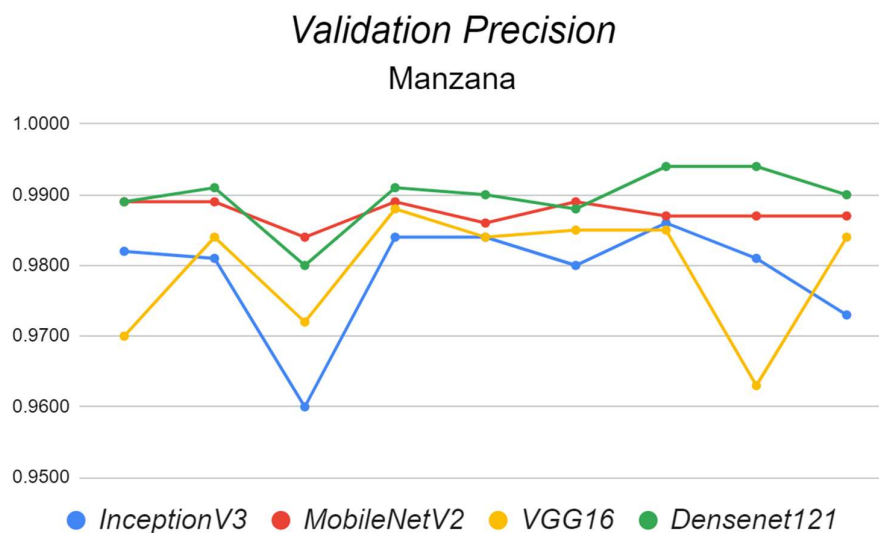


Figura 3.2 *Validation precision* para los entrenamientos con el dataset de imágenes de manzana

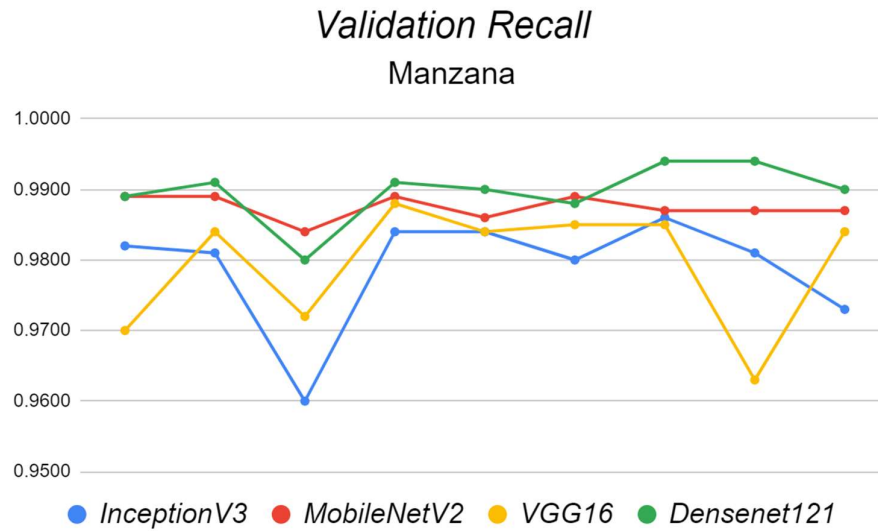


Figura 3.3 Validation recall para los entrenamientos con el dataset de imágenes de manzana

Desde los primeros entrenamientos con 10 épocas estas métricas se mantuvieron por encima del 95%, es decir todos los modelos estudiados logran evitar los falsos positivos y negativos en los resultados con el dataset de validación. Otra métrica que se considera en este análisis es la pérdida o *loss* durante cada iteración de los entrenamientos. Con esta métrica es posible determinar cuán efectiva es la detección de defectos durante el entrenamiento. Según los resultados y como se aprecia en la Figura 3.4 con los entrenamientos de 30 épocas 3 de los 4 modelos lograron reducir esta métrica por debajo de 0.5.

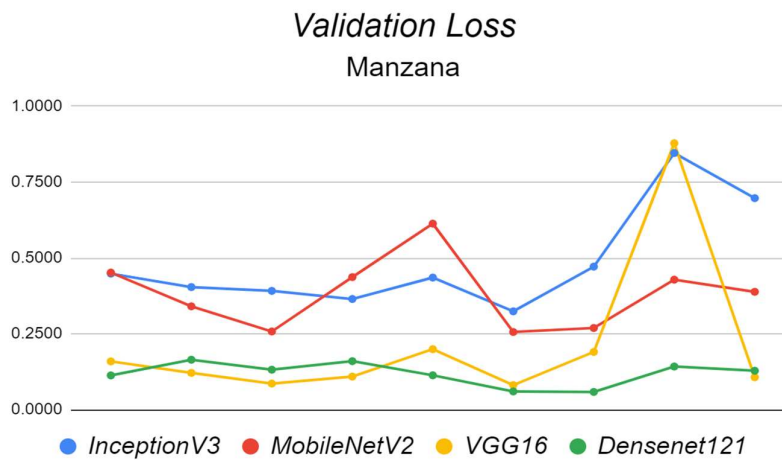


Figura 3.4 Validation loss para los entrenamientos con el dataset de imágenes de manzana

Los valores elevados para la métrica de *loss* indican que existe una gran diferencia entre los valores de los resultados y los valores esperados. En el caso del modelo *InceptionV3* y *MobileNetV2* existe la posibilidad de que exista *overfitting*, dado los valores elevados en la función de *loss* al mismo tiempo que presentan *accuracy* superior al 90%. Este fenómeno de *overfitting* suele ocurrir en modelos densos dado que el paso del gradiente debe ocurrir en toda la red neuronal. Otra causa del *overfitting* al usar *CNN* es la falta de mecanismos de regularización que permitan a la red aprender sin memorizar, una alternativa al *dropout* usado en los entrenamientos de este estudio es el *batch normalization* según [40]. Estos modelos son capaces de reconocer con alto grado de eficacia las imágenes pertenecientes al dataset, pero presentan errores reconociendo imágenes nuevas. Para mitigar este problema con estos modelos se puede realizar entrenamientos con menos épocas y experimentar con más variaciones en la capa de dropout.

3.2 Resultados con el dataset de imágenes de mangos

Para la otra fruta de caso de estudio, el mango, los resultados obtenidos con los modelos seleccionados se asemejan a los obtenidos con la manzana. Los entrenamientos se realizaron usando los mismos hiper-parámetros mencionados anteriormente. En las Tablas 3.5, 3.6, 3.7, 3.8 se puede evidenciar el rendimiento de los modelos para este dataset. Se resalta la fila correspondiente al mejor resultado obtenido en el *test accuracy* para cada modelo.

Tabla 3.5 Métricas de rendimiento de *InceptionV3* para el dataset de imágenes de mangos

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.8145	0.3862	0.9840	0.9840	0.9840	0.8285
10	0.2	0.8335	0.2956	0.9820	0.9820	0.9820	0.8350
10	0.5	0.5665	0.1587	0.9770	0.9770	0.9770	0.5650
20	-	0.9410	0.8218	0.9800	0.9800	0.9800	0.9465
20	0.2	0.8350	0.3915	0.9800	0.9800	0.9800	0.8090
20	0.5	0.8055	0.2793	0.9730	0.9730	0.9730	0.7915
30	-	0.9335	1.2078	0.9780	0.9780	0.9780	0.8710
30	0.2	0.9425	0.7197	0.9870	0.9870	0.9870	0.9295
30	0.5	0.8930	0.4550	0.9830	0.9830	0.9830	0.8875

Tabla 3.6 Métricas de rendimiento de *MobileNetV2* para el dataset de imágenes de mangos

Épocas	<i>Dropout rate</i>	<i>Val Accuracy</i>	<i>Val Loss</i>	<i>Val Precision</i>	<i>Val Recall</i>	<i>F1-score</i>	<i>Test Accuracy</i>
10	-	0.8945	0.4090	0.9860	0.9860	0.9860	0.9030
10	0.2	0.8930	0.2345	0.9840	0.9840	0.9840	0.8790
10	0.5	0.7495	0.0981	0.9890	0.9890	0.9890	0.6900
20	-	0.9410	0.4080	0.9860	0.9860	0.9860	0.9015
20	0.2	0.9535	0.3861	0.9850	0.9850	0.9850	0.9430
20	0.5	0.8850	0.2257	0.9840	0.9840	0.9840	0.8870
30	-	0.8870	0.3543	0.9890	0.9890	0.9890	0.9500
30	0.2	0.9575	0.3647	0.9920	0.9920	0.9920	0.9565
30	0.5	0.9275	0.1851	0.9910	0.9910	0.9910	0.9125

Tabla 3.7 Métricas de rendimiento de VGG16 para el dataset de imágenes de mangos

Épocas	Dropout rate	Val Accuracy	Val Loss	Val Precision	Val Recall	F1-score	Test Accuracy
10	-	0.5290	0.1995	0.9730	0.9730	0.9730	0.5320
10	0.2	0.5125	0.1888	0.9640	0.9640	0.9640	0.5279
10	0.5	0.2680	0.0858	0.9660	0.9660	0.9660	0.2660
20	-	0.6350	0.2157	0.9720	0.9720	0.9720	0.6175
20	0.2	0.7240	0.3417	0.9720	0.9720	0.9720	0.7340
20	0.5	0.4320	0.1250	0.9720	0.9720	0.9720	0.4440
30	-	0.7105	0.2866	0.9800	0.9800	0.9800	0.6835
30	0.2	0.7460	0.2765	0.9740	0.9740	0.9740	0.7495
30	0.5	0.5170	0.1213	0.9770	0.9770	0.9770	0.5305

Tabla 3.8 Métricas de rendimiento de DenseNet121 para el dataset de imágenes de mangos

Épocas	Dropout rate	Val Accuracy	Val Loss	Val Precision	Val Recall	F1-score	Test Accuracy
10	-	0.8565	0.3832	0.9820	0.9820	0.9820	0.8765
10	0.2	0.7550	0.6135	0.9550	0.9550	0.9550	0.7310
10	0.5	0.5385	0.1082	0.9770	0.9770	0.9770	0.5610
20	-	0.9190	0.5408	0.9800	0.9800	0.9800	0.9255
20	0.2	0.8970	0.4640	0.9830	0.9830	0.9830	0.8405
20	0.5	0.7065	0.2401	0.9840	0.9840	0.9840	0.6655
30	-	0.9350	0.5412	0.9840	0.9840	0.9840	0.9300
30	0.2	0.9240	0.5172	0.9820	0.9820	0.9820	0.9280
30	0.5	0.8840	0.3064	0.9850	0.9850	0.9850	0.8975

Los mejores resultados para estos entrenamientos se encontraron con la configuración de 30 épocas y *dropout rate* de 0.2 para los modelos *MobileNetV2* y *VGG16*. Con estos modelos se lograron los valores de 0.9565 y 0.7495 respectivamente. Para los modelos *InceptionV3* y *DenseNet121*, el mejor rendimiento en el dataset de prueba se obtuvo con las configuraciones de 20 y 30 épocas respectivamente, ambas

sin usar capa de *dropout* resultando en *accuracy* de 0.9465 para *InceptionV3* y 0.9300 para *DenseNet121*.

Como se evidencia en la Figura 3.5 para los entrenamientos con las imágenes de mangos se obtuvieron valores de *accuracy* superiores al 75% desde los entrenamientos con 10 épocas.

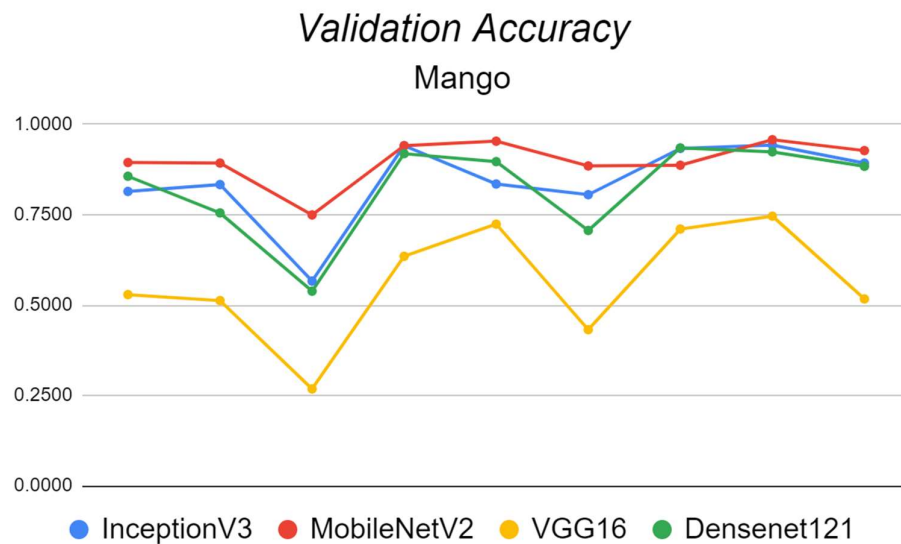


Figura 3.5 Validation accuracy para los entrenamientos con el dataset de imágenes de mango

Al aumentar el número de épocas el *accuracy* resultante se mantiene estable y no varía de forma brusca, excepto con el modelo VGG16. La arquitectura de este modelo es bastante extensa e incluye cientos de millones de parámetros, es posible que se logre una estabilización del *accuracy* obtenido si se entrena con más épocas.

Como se evidencia en las Figuras 3.6 y 3.7, para las métricas de precisión y *recall* ocurre un fenómeno similar que, en los entrenamientos con el dataset de imágenes de manzana, todos los modelos obtuvieron valores superiores al 95% desde las primeras épocas, indicando un alto índice de reconocimiento de las imágenes de entrada y la ausencia de falsos positivos y falsos negativos en los resultados.

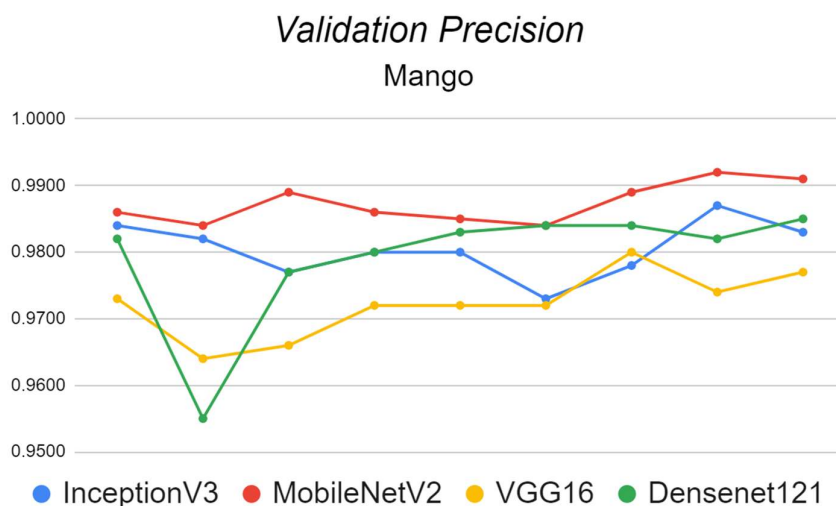


Figura 3.6 *Validation precision* para los entrenamientos con el dataset de imágenes de mango

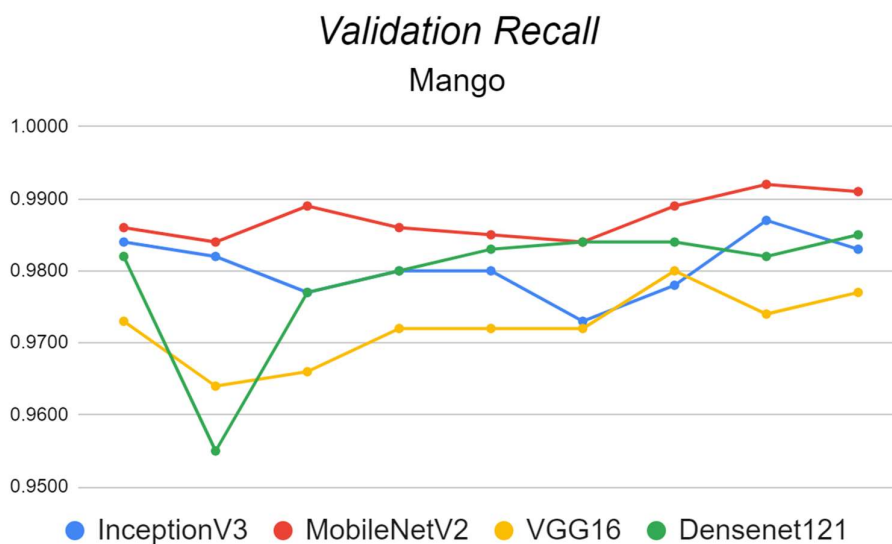


Figura 3.7 *Validation recall* para los entrenamientos con el dataset de imágenes de mango

Para este grupo de entrenamientos los valores de la función de pérdida (*loss*), resultaron más bajos que con el dataset de manzanas. Luego de las 30 épocas, todos los modelos lograron reducir esta métrica a menos de 0.50, y el modelo *VGG16* obteniendo la menor pérdida con sólo 0.1213, como se puede observar en la Figura 3.8.

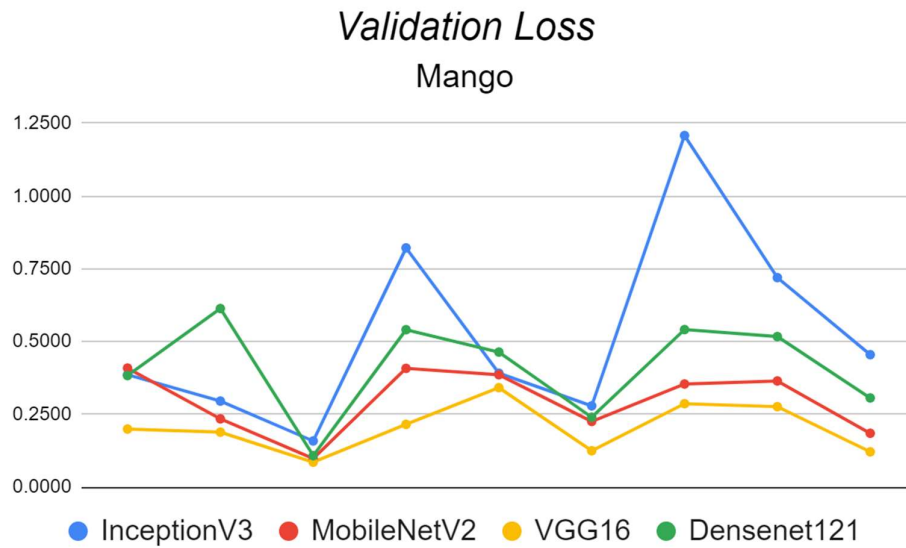


Figura 3.8 Validation loss para los entrenamientos con el dataset de imágenes de mango

En el caso del modelo *InceptionV3*, es posible que exista *overfitting* dado los valores de pérdida superiores a 0.50 obtenidos con 20 y 30 épocas de entrenamiento. Para este modelo en específico se pueden obtener mejores resultados entrenando menos épocas o aplicando variaciones con la capa de *dropout*.

Al usar el prototipo desarrollado para la evaluación de los modelos con imágenes provistas por el usuario se obtienen resultados acordes a las categorías correspondientes de las imágenes. El usuario puede cargar una imagen de la fruta con o sin defectos y seleccionar un modelo de *CNN* para realizar la detección. El prototipo le muestra al usuario el *accuracy* del modelo usado y finalmente muestra el resultado. En las Figuras 3.9, 3.10, 3.11 y 3.12 se pueden observar resultados de la detección realizada por el modelo *MobileNetV2* el cual obtuvo los mejores resultados de *accuracy*. El modelo es capaz de reconocer manzanas y mangos con y sin defectos en diferentes grados.


Results	
Take a look to what we think you have	
Fruit selected	apple
Model selected	mobilenetv2
State	HAS DEFECT
Accuracy	98%
Image provided	

Figura 3.9 Resultado de *MobileNetV2* de manzana con defectos


Results	
Take a look to what we think you have	
Fruit selected	apple
Model selected	mobilenetv2
State	FRESH
Accuracy	98%
Image provided	

Figura 3.10 Resultado de *MobileNetV2* de manzana sin defectos


Results	
Take a look to what we think you have	
Fruit selected	mango
Model selected	mobilenetv2
State	HAS DEFECT
Accuracy	95%
Image provided	

Figura 3.11 Resultado de *MobileNetV2* de mango con defectos


Results	
Take a look to what we think you have	
Fruit selected	mango
Model selected	mobilenetv2
State	FRESH
Accuracy	95%
Image provided	

Figura 3.12 Resultado de *MobileNetV2* de mango sin defectos

En el siguiente capítulo se detallarán las conclusiones y recomendaciones derivadas de los procesos de investigación, recolección de datos, entrenamiento de los modelos y la evaluación de estos. También, se describirán los posibles trabajos futuros en los que se pueden aplicar la metodología descrita en este trabajo, el dataset generado y los modelos entrenados resultado de la experimentación con los hiper-parámetros.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Durante el módulo de creación del *dataset* fue posible recolectar una cantidad considerable de imágenes (20 mil, esto es: 10 mil imágenes para manzanas y 10 mil imágenes para mangos). Este *dataset* contiene imágenes reales e imágenes sintéticas, para el caso de las imágenes reales se usaron bases de datos públicas y para el caso de las imágenes sintéticas se usó el sistema *text-to-image DALL-E mini*.

Dado que la evaluación de modelos de *CNN* para la identificación de defectos es un estudio que ya se ha realizado, como mejora se propuso la compilación de un *dataset* más robusto, no solo en cantidad sino en calidad. El *dataset* que se obtuvo comprende imágenes de frutas sanas y podridas, además de imágenes de defectos en base a las definiciones presentadas de golpes, costras y puntos negros. También, el uso de *DALL-E mini* permitió generar imágenes a conveniencia para los entrenamientos, estas imágenes contienen solo las frutas con los defectos deseados y libres de elementos de fondo y ruido. Cada uno de estos defectos se representó de forma equitativa en el *dataset* para que los modelos logren reconocer otras formas de defectos diferentes a la podredumbre.

En este estudio se provee un *dataset* ordenado por defectos con imágenes de alta calidad para su uso en otros estudios relacionados al control de calidad en manzanas y mangos.

El *dataset* previamente generado fue usado para evaluar 4 modelos de *CNN*, cuyo propósito fue identificar si las frutas en estudio de este proyecto (manzana y mango) son frutas con defectos o frescas. A partir del análisis realizado de las diferentes fuentes literarias referenciadas en este estudio, pudimos extraer 4 modelos *CNN* que destacaron por sus altos valores de exactitud y precisión, los cuales fueron utilizados para diferentes frutas y diferentes defectos, entre ellos, siempre tomando en cuenta las manzanas y mangos, además de los defectos objetivos, tales como: podredumbre, golpes, puntos negros y costras. Los 4 modelos escogidos para este estudio fueron, *MobileNetV2*, *InceptionV3*, *DenseNet121* y *VGG16*, con exactitudes en la etapa de *testing* de: (0.979,

0.956), (0.934, 0.946), (0.932, 0.930) y (0.817, 0.749) respectivamente; los valores en paréntesis corresponden a resultados de “manzana, mango”.

Como resultado de los entrenamientos y la evaluación de los modelos fue posible determinar que *MobileNetV2* es el modelo de *CNN* que mejor se ajusta a la necesidad de clasificación binaria para la detección de defectos en manzanas y mangos. La versatilidad de este modelo lo hace excepcionalmente adecuado para su inclusión en el desarrollo de aplicaciones de control de calidad en frutas, además de ser un modelo ligero y pensado en especial para ejecutarse en dispositivos móviles o con recursos limitados.

Finalmente, fue posible desarrollar un prototipo de aplicación web que permite al usuario realizar detección de defectos usando los modelos entrenados. El prototipo es intuitivo y visualmente llamativo. El usuario carga las imágenes, selecciona el modelo deseado y en poco tiempo obtiene un resultado confiable sobre el estado de la fruta.

4.2 Recomendaciones

Los modelos evaluados en este estudio dan una base comparativa que puede extenderse hacia estudios, tanto de modelos más robustos, como de modelos con arquitecturas reducidas y reconocidos por su rapidez de entrenamiento. Por lo tanto, se recomienda revisar modelos que estén relacionados con el área alimenticia, en especial, el área de control de calidad de frutas y vegetales, para poder determinar si efectivamente los resultados permanecen constantes, con hiper-parámetros distintos e imágenes con mayor calidad.

Se recomienda también, priorizar calidad sobre cantidad, al momento de generar el *dataset*, es preferible tener imágenes de las frutas escenas sin fondo y con el mayor detalle posible, dado que, de esta manera, el modelo se puede concentrar en aprender los factores que caracterizan a la fruta, e ignorar elementos alrededor, como hojas, objetos, personas, e incluso otras frutas que no sean las objetivo.

4.3 Trabajos futuros

Al realizar la evaluación de varios modelos, que tienen precedencia en la literatura de haber sido utilizados en el área de análisis de calidad de alimentos, este estudio puede utilizarse para la elaboración de trabajos con mayor alcance. Por ejemplo, la categorización de defectos por fruta, donde no solo se pueda llevar un conteo de las

frutas en mal estado o en buenas condiciones, sino también, al ser más determinantes con el tipo de defecto encontrado, redirigir el control de calidad a explorar las causas que detonan la aparición de dicho defecto, para de esta manera, elaborar estrategias de prevención de daños, tanto internos, como externos, de la fruta.

Es importante destacar, que este estudio no solo puede verse enfocado en el control de calidad de las frutas, sino también, de cualquier alimento que tenga altos índices de riesgo a ser maltratado o descuidadamente cultivado, de esta forma, se pueden limitar la frecuencia con la que aparecen imperfecciones después de su distribución.

Finalmente, como recomendación para futuros trabajos, si se desea utilizar la misma metodología de creación del dataset, una opción bastante llamativa es, generar las imágenes virtuales, por medio de tomas de modelos 3D o imágenes obtenidas por sensores de medición de superficies, como la tecnología *LiDAR*, que se encuentran incluidas en varias aplicaciones para distintos dispositivos *smart*. Otra herramienta que se pone a consideración es la reciente versión del conversor de texto a imagen *DALL-E 2*, el cual, a diferencia del utilizado en este estudio, puede generar imágenes de más de 1000px, con fondos mostrando un ambiente natural y con una amplia gama de detalles y variedades. Las redes basadas en *transformers* y *GANs* ofrecen la capacidad de generar imágenes detalladas y de alta resolución que pueden mejorar el desempeño de modelos CNN al usar las imágenes generadas para agregar variedad de condiciones y características específicas de los sujetos de estudio en el *dataset* de entrada.

4.4 Contribuciones

- A diferencia de estudios similares donde el los *datasets* son limitados en cuanto a la cantidad de imágenes y la variedad de defectos presentados, se generó un *dataset* robusto de 20 mil imágenes de manzanas y mangos frescos y con defectos. Este dataset contiene imágenes reales e imágenes sintéticas especialmente diseñadas para su uso en la detección de defectos, ya que contienen a la fruta objeto de estudio en su totalidad y sin elementos de fondo.
- En esta investigación se proporcionó una definición de las causas y los defectos más comunes que aparecen en las manzanas y mangos durante las etapas pre y post cosecha.
- Se evaluó un grupo de modelos de *CNN* tomando en cuenta una variedad de

defectos y usando imágenes reales y sintéticas logrando determinar que el modelo *MobileNet* tiene el mejor rendimiento para la solución del problema planteado.

BIBLIOGRAFÍA

- [1] FAO, "Pérdidas y desperdicio de alimentos en el mundo – Alcance, causas y prevención," 2012. [Online]. Available: <https://www.fao.org/3/i2697s/i2697s.pdf>
- [2] B. Zhang *et al.*, "Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review," *Food Research International*, vol. 62, pp. 326–343, 2014, doi: <https://doi.org/10.1016/j.foodres.2014.03.012>.
- [3] J. F. I. Nturambirwe and U. L. Opara, "Machine learning applications to non-destructive defect detection in horticultural products," *Biosystems Engineering*, vol. 189, pp. 60–83, Jan. 2020, doi: [10.1016/j.biosystemseng.2019.11.011](https://doi.org/10.1016/j.biosystemseng.2019.11.011).
- [4] A. Amiri, "Bull's Eye Rot | WSU Tree Fruit | Washington State University," Wsu.edu, 2019. <http://treefruit.wsu.edu/crop-protection/disease-management/bulls-eye-rot/>
- [5] T. Nguyen, "Mango Defect Guide." Accessed: Sep. 03, 2022. [Online]. Available: <http://www.favri.org.vn/attachments/article/490/Mango%20Defect%20Guide.pdf>
- [6] M. Hafiz, "Sala Mango Images Dataset," *Kaggle.com*, 2020. <https://www.kaggle.com/datasets/mypapit/sala-mango-image>.
- [7] R. Koetter and M. Grabowski, "Apple scab of apples and crabapples," *University of Minnesota Extension*, 2019. <https://extension.umn.edu/plant-diseases/apple-scab>.
- [8] J. Crane and R. Gazis, "Bacterial Black Spot (BBS) of Mango in Florida," *UF IFAS Extension*, 2017. <https://edis.ifas.ufl.edu/publication/HS1369>
- [9] A. Omar and Mat, "Principles, methodologies and technologies of fresh fruit quality assurance," *Quality Assurance and Safety of Crops & Foods*, vol. 5, pp. 257–271, Sep. 2013, doi: [10.3920/QAS2012.0175](https://doi.org/10.3920/QAS2012.0175).
- [10] Y. Fu, M. Nguyen, and W. Q. Yan, "Grading Methods for Fruit Freshness Based on Deep Learning," *SN Computer Science*, vol. 3, no. 4, p. 264, 2022, doi: [10.1007/s42979022011527](https://doi.org/10.1007/s42979022011527).
- [11] R. Pathak, "Classification of fruits using convolutional neural network and transfer learning models," *Journal of Management Information and Decision Sciences 1 Journal of Management Information and Decision Sciences*, vol. 24, no. S3, pp. 1–12, 2021, [Online]. Available: <https://www.abacademies.org/articles/classification-of-fruits-using-convolutional-neural-network-and-transfer-learning-models.pdf>
- [12] A. Wu, J. Zhu, and T. Ren, "Detection of apple defect using laser induced light backscattering imaging and convolutional neural network," *Computers & Electrical Engineering*, vol. 81, p. 106454, 2020, doi: <https://doi.org/10.1016/j.compeleceng.2019.106454>.

- [13] E. Houssein, B. Zheng, and T. Huang, "Mango Grading System Based on Optimized Convolutional Neural Network," *Mathematical Problems in Engineering*, vol. 2021, p. 2652487, 2021, doi: 10.1155/2021/2652487.
- [14] J. D. Arango, B. Staar, A. M. Baig, and M. Freitag, "Quality control of apples by means of convolutional neural networks Comparison of bruise detection by color images and near infrared images," *14th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, 1517 July 2020, vol. 99, pp. 290–294, 2021, doi: <https://doi.org/10.1016/j.procir.2021.03.043>.
- [15] M. S. Miah, T. Tasnuva, M. Islam, M. Keya, M. Rahman, and S. Hossain, *An Advanced Method of Identification Fresh and Rotten Fruits using Different Convolutional Neural Networks*. 2021, pp. 1–7. doi: 10.1109/ICCCNT51525.2021.9580117.
- [16] D. Karakaya, O. Ulucan, and M. Turkan, "A Comparative Analysis on Fruit Freshness Classification," *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2019, doi: 10.1109/asyu48272.2019.8946385.
- [17] Ayush Dabra, "Apple Scab Detection Using Deep Learning Based CNN," GitHub, Dec. 06, 2021. <https://github.com/ayushdabra/apple-scab-disease-classification> (accessed Jun. 16, 2022).
- [18] V G, Narendra and A. Pinto, "Defects Detection in Fruits and Vegetables Using Image Processing and Soft Computing Techniques," 2021, pp. 325–337. doi: 10.1007/9789811586033_29.
- [19] A. Wu, J. Zhu, and T. Ren, "Detection of apple defect using laser induced light backscattering imaging and convolutional neural network," *Computers & Electrical Engineering*, vol. 81, p. 106454, 2020, doi: <https://doi.org/10.1016/j.compeleceng.2019.106454>.
- [20] J. L. Charco, A. D. Sappa, B. X. Vintimilla, and H. O. Velesaca, "Camera pose estimation in multi-view environments: From virtual scenarios to the real world," *Image and Vision Computing*, vol. 110, p. 104182, Jun. 2021, doi: 10.1016/j.imavis.2021.104182.
- [21] A. Kolker et al., "The Ray Tracing based Tool for Generation Artificial Images and Neural Network Training," doi: 10.5220/0010168102570264.
- [22] T. Bubeníček and J. Bittner, "Using Game Engine to Generate Synthetic Datasets for Machine Learning." Accessed: Sep. 11, 2022. [Online]. Available: <https://cescg.org/wp-content/uploads/2020/04/Bubenicek-Using-Game-Engine-to-Generate-Synthetic-Datasets-for-Machine-Learning-1.pdf>
- [23] J. Skelton, "DALL-E Mini: Powerful image generation in a tiny model," *Paperspace Blog*, Jul. 20, 2022. <https://blog.paperspace.com/dalle-mini/> (accessed Sep. 11, 2022).
- [24] S. R. Kalluri, "Fruits fresh and rotten for classification," *Kaggle*, 24-Aug-2018. [Online]. Available: <https://www.kaggle.com/datasets/sriramr/fruits-fresh-and-rotten-for-classification>.

- [25] S. Bansal, R. Mehrotra and W. Maffione, "Fruits 360", Kaggle.com, 2021. [Online]. Available: <https://www.kaggle.com/datasets/moltean/fruits>.
- [26] S. Shahane, "Mango varieties classification and grading," Kaggle, 19-Nov-2021. [Online]. Available: <https://www.kaggle.com/datasets/saurabhshahane/mango-varieties-classification>.
- [27] S. Kodors, G. Lacis, O. Sokolova, V. Zhukovs, I. Apeinans and T. Bartulsons. 2021. Apple Scab Detection using CNN and Transfer Learning. *Agronomy Research*, 19(2), 507–519. doi: 10.15159/AR.21.045
- [28] C. G. P. Suescun, "Spoiled and fresh fruit inspection dataset," *Mendeley Data*, 01-Nov-2020. [Online]. Available: <https://data.mendeley.com/datasets/6ps7gtp2wg/1>.
- [29] Naik, Sapan (2019), "Mango Dataset - Studio Setup", Mendeley Data, V1, doi: 10.17632/fmfncxjz3v.1
- [30] Naik, Sapan (2019), "Kesar Mango", Mendeley Data, V1, doi: 10.17632/nsjggt7tyz.1
- [31] S. Nelson, "Mango: Bacterial black spot," Flickr, Sep. 04, 2022. <https://www.flickr.com/photos/scotnelson/albums/72157671996019655> (accessed Sep. 04, 2022).
- [32] K. Shah, "apple fruit disease," Kaggle.com, 2021. <https://www.kaggle.com/datasets/kaivalyashah/apple-disease-detection> (accessed Sep. 04, 2022).
- [33] M. Bednarski, "How DALL-E Mini Works - Towards Data Science," Medium, Jun. 26, 2022. <https://towardsdatascience.com/understanding-how-dall-e-mini-works-114048912b3b> (accessed Sep. 12, 2022).
- [34] S.-H. Tsang, "Review: MobileNetV2 — Light Weight Model (Image Classification)," Medium, May 19, 2019. <https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c> (accessed Sep. 04, 2022).
- [35] S.-H. Tsang, "Review: DenseNet — Dense Convolutional Network (Image Classification)," Medium, Nov. 25, 2018. <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803> (accessed Sep. 04, 2022).
- [36] Adith Narein T, "Inception V3 Model Architecture," OpenGenus IQ: Computing Expertise & Legacy, Sep. 05, 2021. <https://iq.opengenus.org/inception-v3-model-architecture/> (accessed Jul. 14, 2022).
- [37] G. Boesch, "VGG Very Deep Convolutional Networks (VGGNet) - What you need to know - viso.ai," viso.ai, Oct. 06, 2021. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/> (accessed Jul. 14, 2022).

[38] G. Rohini, "Everything you need to know about VGG16 - Great Learning - Medium," Medium, Sep. 23, 2021. <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918> (accessed Jul. 14, 2022).

[39] Tutorial Example, "An Introduction to Accuracy, Precision, Recall & F1-Score in Machine Learning - Machine Learning Tutorial," Tutorial Example, Jan. 11, 2022. <https://www.tutorialexample.com/an-introduction-to-accuracy-precision-recall-f1-score-in-machine-learning-machine-learning-tutorial/> (accessed Sep. 04, 2022).

[40] H. Jansma, "Don't Use Dropout in Convolutional Networks," KDnuggets, 2014. <https://www.kdnuggets.com/2018/09/dropout-convolutional-networks.html> (accessed Sep. 04, 2022).