

# **Sistema Clasificador de Fresa Usando Técnicas de Inteligencia Artificial**

**Juan José Camargo Robles**

Estudiante Facultad de Ingeniería Electrónica



Universidad Santo Tomás  
Facultad de Ingeniería Electrónica  
Tunja, Boyacá, Colombia  
2024

# **Sistema Clasificador de Fresa Usando Técnicas de Inteligencia Artificial**

**Juan José Camargo Robles**

Trabajo de grado presentado como requisito parcial para optar al título de:  
**Ingeniero Electrónico**

Director: Ingeniero Camilo Ernesto Pardo Beainy *Esp. M.sc. PhD(c)*.  
Codirector: Ingeniero Edgar Andrés Gutiérrez Cáceres *Esp. M.sc. PhD(c)*.

Universidad Santo Tomás  
Facultad de Ingeniería Electrónica  
Tunja, Boyacá, Colombia  
2024

## **Exoneración de Responsabilidades.**

Las ideas presentadas en este proyecto son de responsabilidad exclusiva del autor. No es la opinión de la Universidad Santo Tomás o la facultad de Ingeniería Electrónica.

**Aceptación.**

Observaciones:

---

---

---

---

---

\_\_\_\_\_  
**Firma del Decano de la Facultad de Ingeniería Electrónica**

\_\_\_\_\_  
**Firma Primer Jurado**

\_\_\_\_\_  
**Firma Segundo Jurado**

\_\_\_\_\_  
**Firma del Director**

\_\_\_\_\_  
**Firma del Codirector**

## Dedicatoria

Esta investigación representa la culminación de años de esfuerzo, dedicación y perseverancia. Sin embargo, no habría sido posible alcanzar esta meta sin el apoyo incondicional y el aliento constante de personas muy especiales en mi vida. Es por ello que deseo dedicar este logro y fruto de tantos sacrificios, a aquellos seres queridos que siempre creyeron en mí y me motivaron a seguir adelante incluso en los momentos más difíciles. A quien ha sido mi ejemplo toda la vida, mi padre, José Camargo, quien me brindó la oportunidad de dar el primer paso en búsqueda de mis metas, jamás dudó en seguirme apoyando por encima de cualquier situación y por enseñarme el valor del trabajo duro y la perseverancia. A mi madre, Carmen Julia Robles, por ser un eje importante en la construcción de este proceso, por su apoyo incondicional, su sabiduría y su amor en cada paso que daba. A mi hermana, Laura Camargo, por su complicidad, siempre estuvo presente dándome consejos y recomendaciones ayudándome a complementar mi personalidad como mi educación. Al ser más importante en mi vida, mi hijo, Juan Fernando Camargo, que llegó a ser mi motor, a ser el motivo y fuente de inspiración de querer superarme día tras día y así poder luchar para que la vida nos depara un futuro mejor.

Por último, quiero agradecerle a la Familia Moreno Pinzón, quien me abrió la puerta de su hogar brindándome amistad, apoyo y consejos en la etapa más difícil de este camino. A mi amigo e Ingeniero Camilo Ernesto Pardo Beainy, por el apoyo y motivación inquebrantable que me ha brindado desde el primer día que lo conocí. Y todas las personas que han contribuido de una u otra forma en mi formación profesional.

## **Agradecimientos**

Agradezco a la Universidad Santo Tomás seccionales Bucaramanga y Tunja, por brindarme la oportunidad de formarme como profesional e inculcándome valores humanísticos. A mis directores de proyecto, a los ingenieros Camilo Ernesto Pardo Beainy y Edgar Andrés Gutiérrez Cáceres, por su apoyo, orientación, dedicación y liderazgo en el desarrollo de la investigación. A los ingenieros que hacen parte de la facultad de Ingeniería Mecatrónica y Electrónica por inculcarme el anhelo de aprender y la pasión por mi carrera. Y demás personas que contribuyeron indirectamente para que el proyecto sea una realidad.

## Contenido

<b>1. Listas Especiales.....</b>	<b>9</b>
1.1 Lista de Figuras.....	9
1.2 Lista de Tablas. ....	10
<b>2. Resumen.....</b>	<b>11</b>
<b>3. Introducción .....</b>	<b>12</b>
<b>4. Planteamiento del problema .....</b>	<b>14</b>
4.1 Formulación de preguntas de investigación:.....	14
4.2 Definición del problema: .....	14
4.3 Delimitación del problema:.....	14
<b>5. Justificación.....</b>	<b>15</b>
<b>6. Objetivos .....</b>	<b>16</b>
6.1 General:.....	16
6.2 Específicos:.....	16
<b>7. Marco Teórico y Estado del Arte .....</b>	<b>17</b>
7.1 Marco Teórico.....	17
7.1.1 Visión por Computador.....	17
Imagen.....	17
Segmentación.....	18
7.1.2 Machine Learning .....	18
Aprendizaje supervisado (SML).....	19
Aprendizaje no supervisado (UML).....	19
Clustering.....	19
7.1.3 Deep Learning.....	20
Redes neuronales profundas (DNN).....	22
Funciones de activación.....	22
Redes neuronales convolucionales (CNN).....	23
Operación de convolución, en imágenes.....	24
YOLO (You Only Look Once).....	24
YOLOv8.....	25
7.2 Estado del Arte.....	26

<b>8. Desarrollo del proyecto.....</b>	<b>32</b>
8.1 Adquisición y manejo de muestras. ....	32
8.1.1 Aumento de datos. ....	34
8.1.2 Transformaciones.....	34
8.2 Clasificación de las muestras aplicando YOLOv8. ....	36
8.2.1 Entrenamiento del modelo .....	37
8.3 Análisis de datos según la clasificación.....	38
8.4 Determinación del grado de madurez .....	39
8.4.1 Preprocesamiento .....	39
8.4.2 Procesamiento de imagen .....	41
<b>9. Análisis de Resultados y Discusión.....</b>	<b>43</b>
9.1 Clasificación. ....	43
9.2 Detección de nivel de madurez. ....	46
<b>10. Conclusiones y Trabajos Futuros .....</b>	<b>49</b>
<b>11. Referencias Bibliográficas.....</b>	<b>50</b>

## 1. Listas Especiales

### 1.1 Lista de Figuras.

Fig 1. Diagrama de Venn que representa las áreas de la IA y las librerías utilizadas para el desarrollo del proyecto. (Fuente: Autor)	13
Fig 2. Esquema general de visión por computador. La imagen de entrada es procesada para extraer sus características, obteniendo como la salida la descripción de la imagen analizada. (Fuente: Autor)	17
Fig 3. Diagrama de Venn del conjunto de elementos que conforman una imagen. (Fuente: Autor)	18
Fig 4. Ejemplo de segmentación usado para el proceso de separación [11]	18
Fig 5. Diferencias entre la programación tradicional y Machine Learning. (Fuente: Autor)	19
Fig 6. Ejemplo de clustering. (Fuente: Autor)	20
Fig 7. Ejemplo del cálculo de centroides [15]	20
Fig 8. Red neuronal profunda para la clasificación de tres flores. (Fuente: Autor)	21
Fig 9. Arquitectura de red neuronal. (Fuente: Autor)	21
Fig 10. Arquitectura de YOLOv8. (Fuente: Autor)	25
Fig 11. Arquitectura de trabajo llevado a cabo para la clasificación y determinación de madurez de fresa. (Fuente: Autor)	32
Fig 12. Taxonomía de los enfoques de aumentos de datos en esta investigación. (Fuente: Autor)	34
Fig 13. Interfaz de etiquetado de Roboflow. (Fuente: Autor)	35
Fig 14. Modelos utilizados para el aumento de datos del dataset original. (Fuente: Autor)	35
Fig 15. Matriz de confusión. (Fuente: Autor)	38
Fig 16. Fresa en condiciones ideales. (Fuente: Autor)	39
Fig 17. Mascara antes y después de realizadas las operaciones morfológicas. (Fuente: Autor)	40
Fig 18. Pérdidas de entrenamiento del primer modelo. (Fuente: Autor)	43
Fig 19. Métrica de entrenamiento del modelo con data augmentation. (Fuente: Autor)	44
Fig 20. (a) Curva de aprendizaje del primer modelo, (b) Curva de aprendizaje del modelo con data augmentation. (Fuente: Autor)	44
Fig 21. Metricas graficas de la evaluación de los modelos antes y después del aumento de datos respectivamente. (Fuente: Autor)	45
Fig 22. Imagen original en Espacio de Color HSV. (Fuente: Autor)	46
Fig 23. Histograma de clústers en imagen HSV. (Fuente: Autor)	46
Fig 24. Máscara de imagen original. (Fuente: Autor)	47
Fig 25. Imagen original segmentada. (Fuente: Autor)	47
Fig 26. Imagen segmentada transformada en espacio de color LAB. (Fuente: Autor)	48
Fig 27. Clusterización de la imagen LAB. (Fuente: Autor)	48
Fig 28. Porcentajes de inmadurez del fruto. (Fuente: Autor)	48

## 1.2 Lista de Tablas.

Tabla 1. Funciones de activación más comunes. (Fuente: Autor) .....	23
Tabla 2. Técnicas de Inteligencia Artificial para clasificación y segmentación de imágenes. (Fuente: Autor) .....	26
Tabla 3. Conjunto de muestras tomadas, para el entrenamiento de YOLOv8. En la parte inferior se pueden observar muestras en conjunto de datos. (Fuente: Autor).....	33
Tabla 4. Construcción y distribución del dataset obtenido. (Fuente Autor).....	36
Tabla 5. Modelos de YOLOv8 usados para el entrenamiento. (Fuente: Autor).....	36
Tabla 6. Pseudocódigo de Clasificación. (Fuente: Autor) .....	37
Tabla 7. Pseudocódigo de Determinación de Madurez (Primera Parte). (Fuente: Autor).....	40
Tabla 8. Pseudocódigo de Determinación de Madurez (Segunda Parte). (Fuente: Autor).....	42
Tabla 9. Evaluación de métricas del modelo antes y después del aumento de datos. (Fuente: Autor) .....	45

## 2. Resumen

En el desarrollo de esta investigación, se implementan varios algoritmos de inteligencia artificial para la clasificación y detección del nivel de madurez de fresa. Las imágenes adquiridas por el sistema de sensado de una banda transportadora, fueron sometidas, por una parte; a algoritmos de Machine Learning y Deep Learning, orientados a la clasificación del fruto en dos categorías, definidas por sus características de madurez y deficiencias nutricionales (afectaciones visuales), y, por otra parte; a algoritmos de Deep Learning, orientados a la extracción de características para determinar su nivel de madurez.

Se creó un dataset público desde cero con el objetivo de que esté disponible para cualquiera que lo necesite. Para alcanzar una clasificación satisfactoria, se entrenaron modelos utilizando la arquitectura YOLO, específicamente la versión ocho, lo que permitió realizar la clasificación del fruto con éxito. Posteriormente, se llevaron a cabo pruebas sobre las imágenes previamente clasificadas, empleando técnicas como segmentación de fondo por umbrales y segmentación cromática. Estas técnicas de ingeniería de características facilitaron la detección del nivel de madurez del fruto.

## SISTEMA CLASIFICADOR DE FRESA USANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL

### 3. Introducción

En la actualidad, las nuevas tecnologías han transformado significativamente diversos sectores, incluyendo la agricultura, donde han influido en la explotación agraria y la comercialización de productos agroalimentarios. En particular, la inteligencia artificial en el ámbito agrícola se emplea para aumentar la eficiencia y precisión en las operaciones, contribuyendo así a la sostenibilidad y asegurando la calidad de los productos.

La aplicación de la Inteligencia Artificial (IA) en la agricultura es un campo de alta relevancia y amplia trayectoria, dado que aproximadamente el 30.7% de la población mundial está directamente involucrada en la gestión de 2781 millones de hectáreas de terreno agrícola. Esta integración de la IA en la agricultura implica la recopilación y análisis de datos provenientes de diversas fuentes, tales como sensores, maquinaria agrícola, imágenes satelitales y drones, dispositivos móviles, los cuales son transformados en información útil. Este enfoque tecnológico ha demostrado ser beneficioso para múltiples aspectos del sector agrícola, incluyendo el manejo general de cultivos, control de plagas, gestión de suelos, supervisión y control de la producción, predicción de rendimientos, y generación de información para análisis de sostenibilidad, entre otras aplicaciones.

Los primeros registros de la aplicación de la Inteligencia Artificial en la agricultura datan de 1983 [1]. Desde entonces, se han propuesto diversos enfoques para abordar los desafíos inherentes a este sector, que van desde técnicas de Internet de las Cosas (IoT) [2] hasta sistemas de apoyo a la toma de decisiones basados en Aprendizaje Automático [3].

Teniendo en cuenta que la inteligencia artificial y las áreas que serán de estudio en este libro (Fig 1), están fundamentada en el análisis estadístico, el cálculo diferencial – multivariado, álgebra lineal, entre otras áreas, fueron desarrolladas previamente en las máquinas conocidas como computadores de propósito general, así mismo los planteamientos de la IA, fueron previos al auge comercial de la computación que se vive actualmente. Las ideas como el perceptrón, la retro propagación y el test de Turing, habían sido establecidas y formuladas, antes de que la industria de los videojuegos impulsara el desarrollo tecnológico de los componentes en los cuales hoy en día se hace el procesamiento de los algoritmos de IA, estos módulos se refieren a la CPU y GPU.

El primer paso en el desarrollo de este proyecto de investigación en el ámbito de la Inteligencia Artificial se origina en una de sus ramas especializadas conocida como Deep Learning. En esta fase, se propone la utilización de una biblioteca ampliamente reconocida en el campo del aprendizaje automático, que permite la clasificación de un conjunto de imágenes previamente adquiridas mediante un sistema de sensores. Posteriormente, estas imágenes son sometidas a una serie de técnicas diseñadas para extraer y resaltar características relevantes. Los fundamentos de la investigación se delinearán con la consideración de que, desde esta perspectiva, es posible abordar diversas problemáticas que implican la intervención humana.

Este documento expone una solución que abarca desde la adquisición de imágenes hasta el análisis de datos, garantizando un monitoreo continuo y confiable para la toma de decisiones en la

etapa postcosecha. Se detallan los desafíos específicos que aborda la propuesta, las soluciones técnicas implementadas, y los beneficios esperados de su aplicación en el contexto agrícola.

Para que una tarea de aprendizaje automático funcione correctamente se deben de establecer tres lineamientos que son fundamentales al momento de dar solución [4], basándonos de la definición de aprendizaje de maquina: “Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P, si su desempeño en las tareas en T, medido por P, mejora con la experiencia E”, para el caso de la investigación como se divide en dos partes, la primera corresponde: Tarea (T: Clasificar), medida de desempeño (P: Precisión) y la experiencia (E: Aprendizaje supervisado); dándonos a la segunda parte: Tarea (T: Segmentación), medida de desempeño (P: Precisión) y la experiencia (E: Aprendizaje no supervisado).

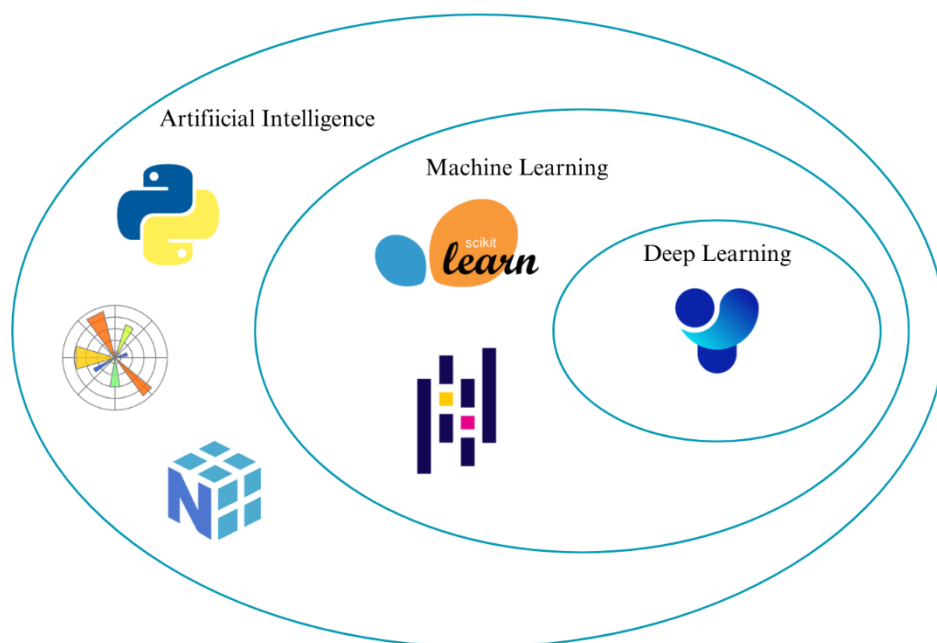


Fig 1. Diagrama de Venn que representa las áreas de la IA y las librerías utilizadas para el desarrollo del proyecto. (Fuente: Autor)

## **4. Planteamiento del problema**

### **4.1 Formulación de preguntas de investigación:**

¿Cuáles son algunas estrategias y tecnología recomendadas para mejorar la eficiencia y precisión en las etapas posteriores a la cosecha y clasificación de fresas, con el objetivo de elevar la calidad del producto y fortalecer la competitividad de los productores en la industria de las fresas?

### **4.2 Definición del problema:**

A raíz de una investigación realizada en la Universidad Santo Tomás de Aquino, se ha detectado la necesidad de diseñar un sistema automatizado que mejore la eficiencia y precisión en los procedimientos postcosecha y de clasificación de fresas. Este sistema busca garantizar la calidad del producto final y elevar la competitividad de los productores de fresas en Colombia tanto a nivel nacional como internacional. Específicamente, se enfoca en las exportaciones, siendo Colombia el tercer país latinoamericano con mayor área sembrada de cultivos frutales entre ellos la fresa [5], lo que supone una importante contribución a la economía del país.

### **4.3 Delimitación del problema:**

El problema previamente delineado se centra en abordar los desafíos específicos enfrentados por los productores de fresas en Colombia, con una especial atención en la provincia Centro de Boyacá. Se tomarán en consideración las características de madurez de la fresa presentes en esta región, particularmente su color, como punto de partida para la implementación de un sistema de clasificación. Se establecerán tres categorías con atributos definidos, que servirán como estándares en el proceso de clasificación.

El objetivo de este proyecto es desarrollar un prototipo de selección de fresas utilizando técnicas de inteligencia artificial. Para garantizar la eficacia y relevancia del algoritmo de clasificación, se llevarán a cabo pruebas para validar su capacidad para clasificar las fresas de manera precisa y eficiente. Además, se evaluará la solución propuesta en relación con los desafíos identificados en el contexto agrícola de Boyacá.

## 5. Justificación

En Colombia, la actividad agrícola relacionada con el cultivo de manazas, duraznos, peras, ciruelas y fresas posee una extensa trayectoria y actualmente contribuye con el 19% de la producción total de frutas en el país. Según la Dirección de Cadenas Agrícolas y Forestales (Marzo de 2021), en el año 2020, Boyacá reportó una producción de 136.125 toneladas de frutas las cuales 3.407 toneladas correspondieron a la fresa [5]. Es relevante señalar que durante el periodo de 2019 a 2021, se observó un crecimiento sostenido en la producción por hectárea. En la actualidad, las fresas dominan el mercado y resultan un proceso de siembra distribuido entre los municipios de Tuta con 50 hectáreas, Toca con 28 hectáreas y Oicatá con 16 hectáreas aproximadamente [6], desempeñando un papel crucial en la producción frutícola de Boyacá.

En la provincia Centro de Boyacá, el cultivo de frutas como la uchuva, gulupa, pitahaya, fresa y tomate de árbol constituye una actividad económica primordial que carece de avances tecnológicos adecuados. Las labores de producción, especialmente la selección de fresas, son altamente intensivas en mano de obra y tiempo, lo que resulta en retrasos significativos en la comercialización y costos elevados debido a la necesidad de contratar gran cantidad de trabajadores.

La fluctuación de precios de mercado, que a menudo no cubren los costos de producción, junto con la congestión en el mercado debido al exceso de oferta en ciertas temporadas, contribuye a almacenamientos prolongados que aceleran la maduración y descomposición de los productos.

Este proyecto se enfoca en mejorar la eficiencia económica del agricultor al abordar específicamente el problema de la selección de fresas, lo que reduciría los costos y el tiempo dedicado a esta tarea. La modernización de este proceso permitirá reasignar recursos a otras actividades fundamentales del cultivo. Además, se consideran aspectos importantes que suelen pasarse por alto durante las jornadas de trabajo intensivas.

## 6. Objetivos

### 6.1 General:

Desarrollar un sistema de clasificación de fresa según sus características de madurez ayudado de técnicas de Inteligencia Artificial.

### 6.2 Específicos:

- Determinar los parámetros visuales necesarios para establecer grados de madurez homogeneizada en el fruto y los procedimientos de sensado relacionados.
- Seleccionar las técnicas de Inteligencia Artificial para desarrollar el proceso de selección de fresa.
- Implementar un algoritmo de aprendizaje automático, que, aprovechando los parámetros visuales del procesamiento de imágenes, permita la clasificación de fresa en diferentes categorías.

## 7. Marco Teórico y Estado del Arte

### 7.1 Marco Teórico.

#### 7.1.1 Visión por Computador

La visión artificial, también conocida como visión por computadora, ha experimentado un aumento significativo en su uso en los últimos años y a menudo se confunde con otros conceptos relacionados. Esta disciplina se considera una subrama de la inteligencia artificial clásica y abarca una serie de fases necesarias para dotar a las computadoras de la capacidad de percibir y comprender imágenes, imitando así el proceso realizado por los seres humanos. Aunque la arquitectura y las etapas de un sistema de visión por computadora pueden variar según la aplicación específica, hay varias etapas comunes que se encuentran en la mayoría de las implementaciones. Estas etapas incluyen la adaptación de la imagen, el procesamiento, la segmentación, la representación y descripción, así como el reconocimiento e interpretación de la imagen [7].

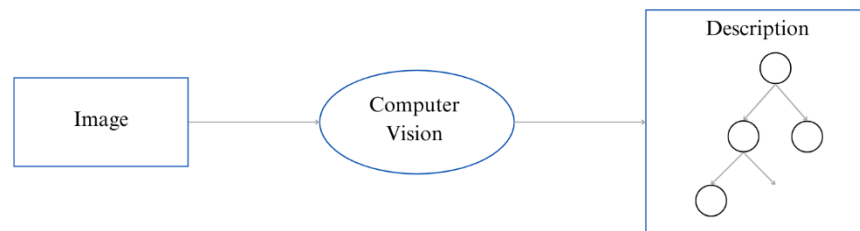


Fig 2. Esquema general de visión por computador: La imagen de entrada es procesada para extraer sus características, obteniendo como la salida la descripción de la imagen analizada. (Fuente: Autor)

**Imagen.** La imagen se puede definir como una representación bidimensional de una escena en un mundo tridimensional. Esta imagen se obtiene a partir de una señal capturada por un sensor, el cual convierte la información del espectro electromagnético en codificaciones numéricas. Por lo general, una imagen digital se transforma en un formato específico de representación de imágenes y está definida por un arreglo de datos discretos tanto en los valores que pueden adoptar como en los parámetros que la definen. Normalmente, una imagen digital se representa como una matriz de dimensiones  $(n \times m)$ , donde cada elemento de la matriz contiene un valor discreto que cuantifica el nivel de información correspondiente [8]. Así, una imagen puede formalizarse como una función discreta de dos dimensiones de la siguiente manera:

$$G(x, y) / 0 \leq x \leq n - 1, 0 \leq y \leq m - 1$$

Una imagen se representa como una función bidimensional que proporciona información electromagnética para cada uno de sus valores discretos, denominados como puntos o píxeles. Cada píxel contiene información sobre el nivel de iluminación o color correspondiente a un punto específico en la escena. Esta representación espectral es captada por un sensor. Por ejemplo, en una imagen RGB, el canal verde corresponde a la representación espectral de ciertas longitudes de onda, generando el color mediante la superposición de tres componentes espectrales.

La formación de una imagen se produce cuando el sensor de la cámara registra la radiación que ha interactuado con los objetos físicos presentes en la escena. El valor de la función en cada píxel de la imagen corresponde a la intensidad o brillantez de la radiación captada [9]. Una vez

capturada, en la imagen se produce una transducción eléctrica, que es posteriormente discretizada y almacenada en forma de imagen digital.

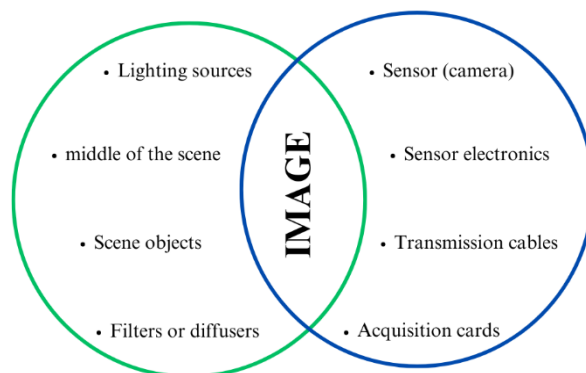


Fig 3. Diagrama de Venn del conjunto de elementos que conforman una imagen. (Fuente: Autor)

**Segmentación.** El objetivo principal de la segmentación de imágenes es dividir una imagen en regiones, cada una de las cuales presenta una apariencia visual homogénea o corresponde a objetos o partes de objetos [10]. En este proceso, cada píxel de la imagen se representa como un punto en un espacio tridimensional compuesto por las intensidades de los canales rojo, verde y azul. Es importante considerar que cada píxel es independiente y que este espacio no es euclidiano debido a que las intensidades de los canales están limitadas.

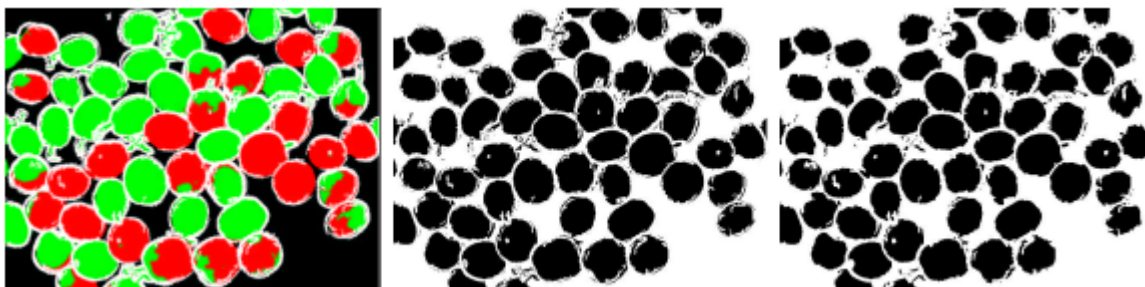


Fig 4. Ejemplo de segmentación usado para el proceso de separación [11]

### 7.1.2 Machine Learning

El aprendizaje automático (Machine Learning) consiste en el estudio de algoritmos y modelos que tienen la capacidad de aprender de los datos disponibles y realizar tareas específicas sin requerir una programación explícita para cada una de ellas. Este enfoque se basa en la premisa fundamental de que los sistemas pueden mejorar su desempeño a través de la experiencia, es decir, al analizar y procesar datos de forma iterativa [12]. Los algoritmos de ML ofrecen una alternativa innovadora al paradigma de la programación tradicional. En este último, las reglas y los datos se utilizan como entradas para producir un resultado predeterminado. En contraste, en el aprendizaje automático, los datos mismos constituyen las entradas junto con los resultados (especialmente cuando las clases están etiquetadas), también se emplean como datos de entrada. La salida de estos algoritmos son reglas o modelos que describen y predicen los datos (Fig 5). Este enfoque flexible y adaptable proporciona una poderosa herramienta para resolver una amplia gama de problemas en diversos campos y aplicaciones.

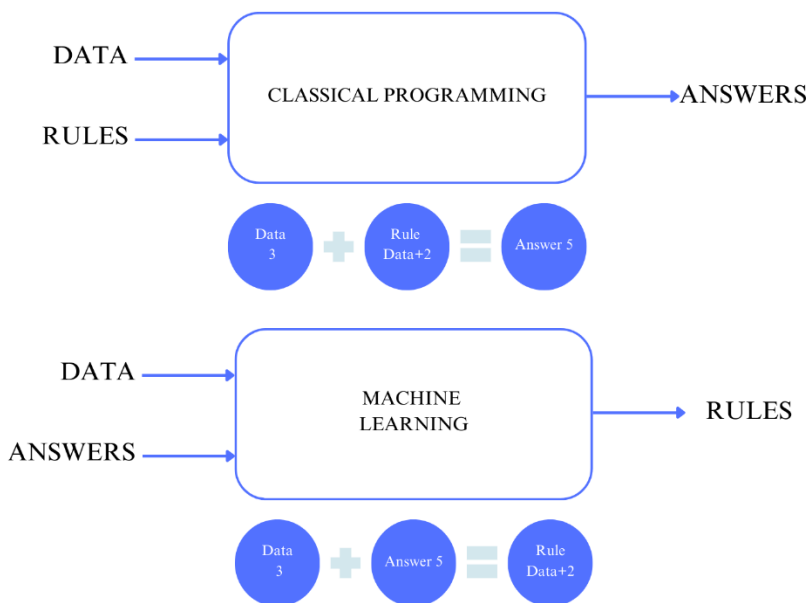


Fig 5. Diferencias entre la programación tradicional y Machine Learning. (Fuente: Autor)

Dentro de la disciplina del Machine Learning, se distinguen diversos tipos de aprendizaje que se basan en la experiencia del algoritmo. Estos incluyen el aprendizaje supervisado, no supervisado, auto supervisado y de refuerzo. En el contexto de este documento, se enfoca específicamente en el aprendizaje supervisado y en el no supervisado.

**Aprendizaje supervisado (SML).** El aprendizaje supervisado en Machine Learning, es un modelo el cual se entrena utilizando ejemplos de entrada y salida, con el objetivo de aprender una función que pueda mapear de manera precisa las entradas y salidas correspondientes, resaltando que esto solo se aplica cuando cada dato, o conjunto de datos de entrada (muestra) tiene asociada una etiqueta [13], por ejemplo: un dataset de imágenes en las que cada una de ellas contiene algún tipo de metadato (normalmente una etiqueta o conjunto de etiquetas): “((imag0001.bmp, ‘gato’), (imag0002.bmp, ‘café’), (imag0003.bmp, ‘flor’). Este enfoque permite al modelo realizar predicciones precisas sobre nuevos conjuntos de datos. La mayoría de las aplicaciones de Deep Learning, que actualmente reciben gran atención, pertenecen a esta categoría. Este tipo de aprendizaje se centra en problemas de clasificación y regresión.

**Aprendizaje no supervisado (UML).** En diferencia al aprendizaje supervisado, consiste en entrenar un modelo sin proporcionar una variable de respuesta, permitiendo que el algoritmo se centre en la detección de estructuras, relaciones, tendencias, conglomerados y/o atípicos datos en la entrada, este proceso se le conoce comúnmente como data mining descriptivo.

**Clustering.** Este algoritmo se basa en agrupar un conjunto de datos en subconjuntos llamados clústers, de modo que los datos dentro de cada clúster sean más similares entre sí que a los datos de otros clústers. En estadística, este problema se conoce como análisis de conglomerados y es un área ampliamente estudiada con una gran variedad de algoritmos desarrollados. Por lo tanto, se puede entender el algoritmo como un generador de un vector de código one-hot de  $k$ -dimensiones,  $h$ , que representa una entrada  $x$ ; si  $x$  pertenece al clúster  $i$ , entonces  $h_i = 1$  y todos

los demás elementos del vector  $h$  son cero. En el siguiente gráfico (Fig 6), se muestra un esquema típico de clustering.

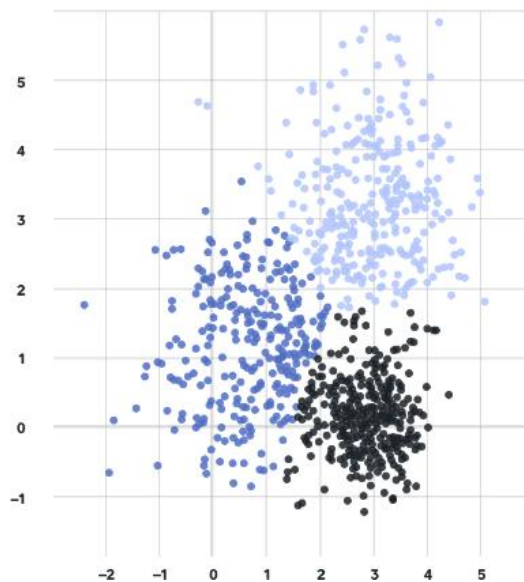


Fig 6. Ejemplo de clustering. (Fuente: Autor)

Un modelo de clustering podría proporcionar varios hiperplanos lineales de separación, mientras que un modelo de clustering diferente podría proporcionar algunos elementos virtuales representativos: *centroides*, cuya “área de influencia” determina a que clúster pertenece cada una de las muestras, en base al contexto del documento, el algoritmo *k-means* comienza inicializando  $k$  centroides  $\{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(k)}\}$  con valores aleatorios. Luego, alterna entre dos pasos hasta converger: primero asigna cada ejemplo de entrenamiento al clúster del centroide más cercano; segundo, actualiza cada centroide calculando la media de los ejemplos asignados a su clúster. Este proceso se repite iterativamente hasta que los centroides dejan de cambiar significativamente, indicando que los clústers están bien definidos [14]. El siguiente grafico (Fig 7) muestra un ejemplo de evolución de centroides, de cara a ajustar los clústers pertinentes según la situación presente.

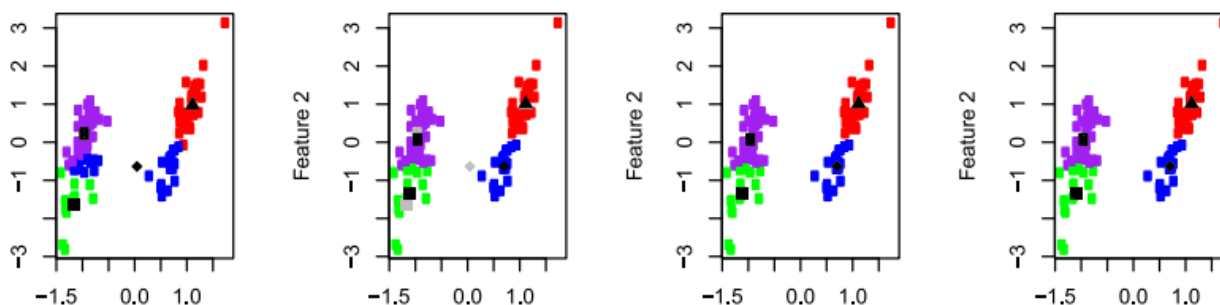


Fig 7. Ejemplo del cálculo de centroides [15]

### 7.1.3 Deep Learning

El aprendizaje profundo es un subcampo del aprendizaje automático (Fig 1), es decir, una versión actualizada del aprendizaje en representaciones; en contraste con los métodos de aprendizaje automático tradicionales, donde los ingenieros a menudo deben diseñar y seleccionar manualmente las características relevantes para una tarea específica, el Deep Learning se enfoca

en aprender representaciones de datos en múltiples niveles de abstracción. Estas representaciones se obtienen a través de la composición de múltiples transformaciones no lineales, generalmente en forma de redes neuronales profundas (DNN), las cuales usan el término “Deep” por las múltiples capas de unidades de procesamiento, lo que les permite aprender de manera jerárquica características cada vez más abstractas de los datos de entrada, permitiendo que las características relevantes se aprendan automáticamente a partir de los datos, esto se logra mediante el *backpropagation* y el *gradient descent*, que permiten ajustar los parámetros de la red para minimizar alguna función perdida.

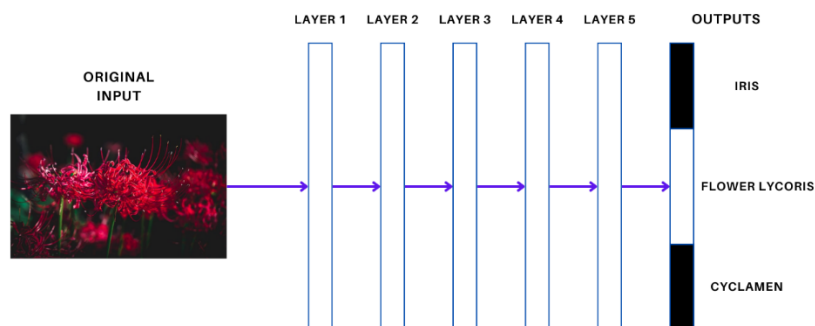


Fig 8. Red neuronal profunda para la clasificación de tres flores. (Fuente: Autor)

En la Fig 8 se observa como una imagen que representa datos de entrada se transforma a través de la arquitectura de la red, para determinar a qué tipo de flor pertenece la imagen de entrada [14].

Este subcampo de la inteligencia artificial se centra en las redes neuronales, que se basan en la interconexión de nodos y en la cantidad de capas presentes. La arquitectura de una red neuronal se define por cómo se conectan estos nodos y por los niveles de nodos entre la entrada y la salida, así como la cantidad de neuronas en cada capa (ver Fig 9), lo cual es crucial para su capacidad de aprendizaje y entramiento.

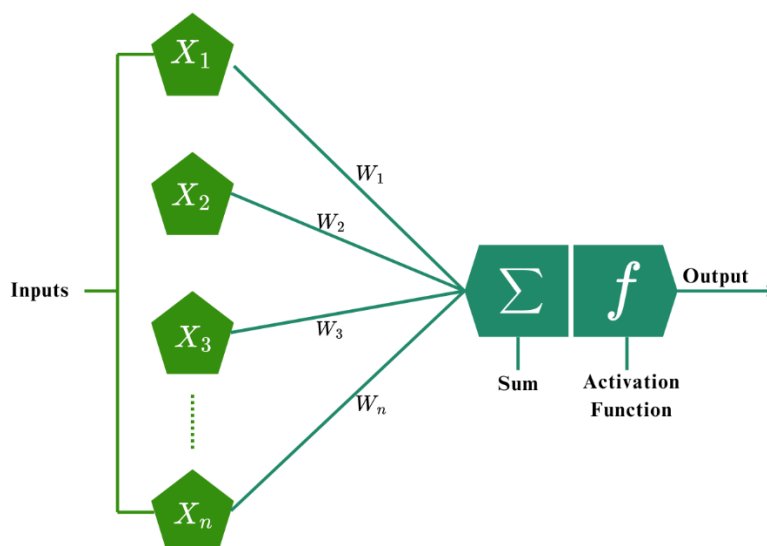


Fig 9. Arquitectura de red neuronal. (Fuente: Autor)

Existen diversos tipos de arquitecturas en redes neuronales, que se pueden clasificar en cuatro grupos principales: redes neuronales profundas (DNN), redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y arquitecturas emergentes (EAS). En la siguiente sección de este capítulo, se centrará en un análisis más detallado de las DNN y las CNN.

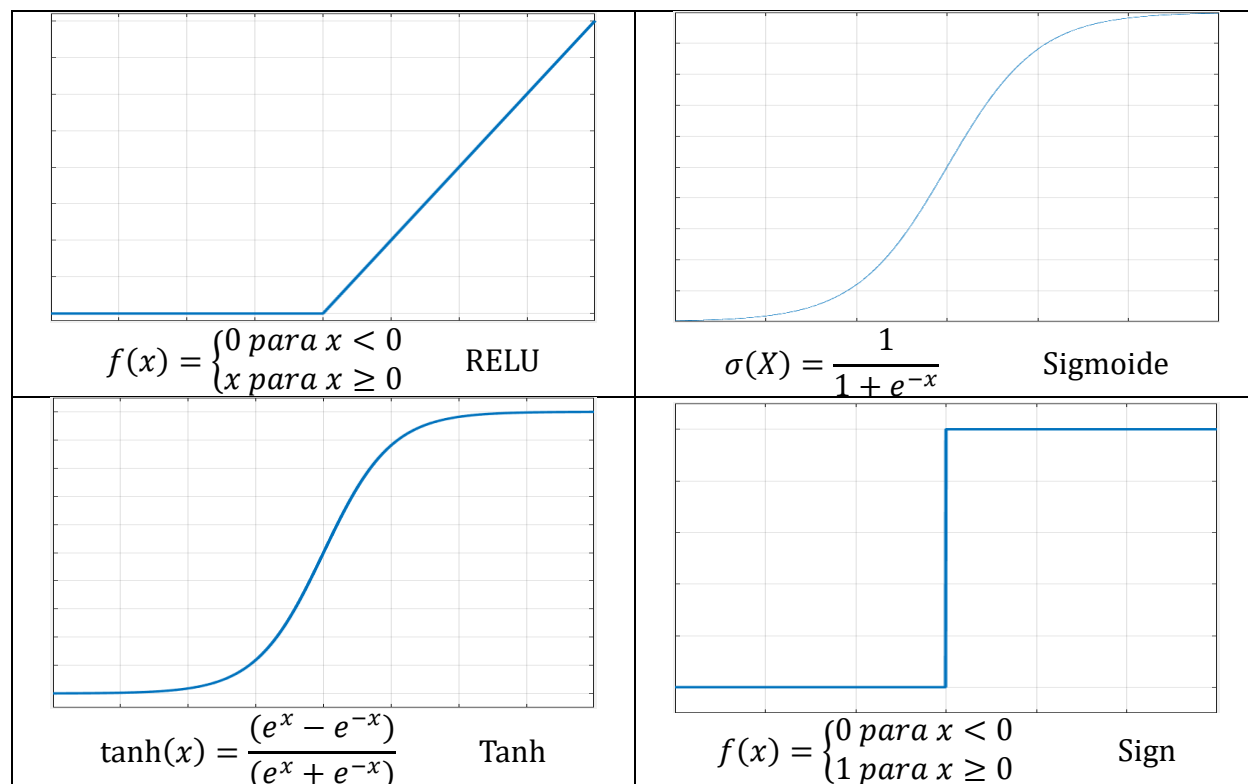
**Redes neuronales profundas (DNN).** Las redes neuronales profundas son redes neuronales artificiales que contienen múltiples capas ocultas entre la capa de entrada y la capa de salida. Estas redes son especialmente útiles cuando los procedimientos de análisis tradicionales no son aplicables debido a la complejidad de los datos que deben procesarse, ya que permiten modelar esta complejidad de manera efectiva [16]. En otras palabras, la capacidad de aprendizaje de la red aumenta con su profundidad. Un perceptrón (unidad fundamental de una red neuronal, consta de una sola capa de entrada y un nodo de salida), realiza la suma ponderada de sus entradas, lo que representa una función lineal. Sin embargo, al tener múltiples capas con múltiples neuronas, la combinación de estos comportamientos lineales puede representar funciones mucho más complejas. Una red neuronal profunda se define como una composición de funciones lineales y no lineales, donde cada capa oculta introduce una transformación adicional que permite capturar patrones complejos en los datos, formalmente se define así.

$$\hat{y} = \sum_{k=1}^n W_n A_n + B_n$$

Se puede deducir que la función de salida de una red con múltiples capas, que no posee una función de activación (se reduce al funcionamiento de una regresión lineal), lleva a la suma ponderada de las regresiones lineales conectadas a la capa de salida, lo que resulta en la combinación lineal de las capas anteriores, donde  $n$  representa la capa y  $A_0 = X$ . Debido a este tipo de arquitectura, una red neuronal sin funciones de activación se reduciría a una sola neurona conectando directamente la capa de entrada a la de salida. Esto significa que, sin funciones de activación, una red neuronal densamente conectada no puede capturar la complejidad necesaria para modelar relaciones no lineales en los datos.

**Funciones de activación.** Sin una función de activación, una capa densa se reduciría a solo dos operaciones lineales: un producto escalar y una suma. En Deep Learning, las funciones de activación son imprescindibles porque introducen no linealidades en las neuronas de cada capa. Por ejemplo, en un perceptrón se utiliza una función de activación llamada *sign*. Estas funciones permiten apilar una gran cantidad de neuronas de manera secuencial. Sin funciones de activación, una capa solo podría aprender transformaciones lineales, lo que es insuficiente para modelar conjuntos de datos complejos. Para lograr un aprendizaje más robusto y capturar patrones complejos en los datos, es esencial utilizar funciones de activación no lineales [14].

Tabla 1. Funciones de activación más comunes. (Fuente: Autor)



En la **¡Error! No se encuentra el origen de la referencia.**, se presentan las funciones de activación más conocidas junto con sus respectivos modelos matemáticos. Por ejemplo, la función sigmoide convierte los valores grandes que ingresan a la neurona en valores cercanos a uno, mientras que los valores pequeños se saturan a cero. De manera similar, la función tangente hiperbólica satura los valores pequeños a -1 y los valores grandes a 1. La elección de estas funciones de activación depende directamente del tipo de tarea y de la función objetivo que se desea alcanzar, ya que diferentes funciones pueden influir significativamente en el rendimiento del modelo en distintas aplicaciones.

**Redes neuronales convolucionales (CNN).** Estas son una variante de las redes neuronales profundas (DNN). Se especializan específicamente en el procesamiento de datos estructurados de forma de cuadrícula, como imágenes y señales de audio. Estas son particularmente efectivas en el procesamiento de datos con una estructura espacial como lo son las imágenes, debido a su capacidad de capturar patrones locales y espaciales de manera eficiente. Esta emplea la operación matérica de la convolución, donde la convolución es una operación lineal especializada y estas aprovechan para detectar características locales como bordes, texturas y patrones visuales simples. Además, las capas de submuestreo (o pooling) se utilizan para reducir la dimensionalidad de los mapas de características, preservando las características más importantes [12]. Esto reduce el riesgo de sobreajuste y requiere muchos menos datos de entrenamiento. Además, cuando una CNN ha aprendido un kernel que puede detectar una característica particular, puede detectar esa característica en cualquier parte de la imagen.

**Operación de convolución, en imágenes.** Esta convolución es una operación matemática que combina dos funciones para producir una tercera. En el contexto del procesamiento de imágenes y redes neuronales convolucionales, esta operación se utiliza para aplicar un filtro (kernel) a una entrada para extraer las características específicas, como bordes, texturas o patrones, esto sucede cuando el kernel, se desliza a través de una imagen, generando una imagen de salida con un alto valor de dimensionalidad.

En aplicaciones de Machine Learning, trabajamos frecuentemente con matrices multidimensionales llamadas tensores, que permiten almacenar cada elemento de entrada de manera explícita y separada. La convolución es una operación clave en estas aplicaciones, donde se realiza una suma de los productos entre los valores de entrada y un kernel. Este kernel es una matriz multidimensional de parámetros que se ajusta durante el proceso de aprendizaje del algoritmo. En otras palabras, los filtros convolucionales en estas redes se convierten en parámetros manejables que permiten capturar características importantes de los datos de entrada, como patrones visuales en imágenes o secuencias en datos temporales, facilitando así el aprendizaje y la extracción de características relevantes para la tarea en cuestión.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a)$$

A medida que la dimensionalidad de la imagen aumenta, el kernel debe irse adaptando, normalmente se implementa la función llamada correlación cruzada, que es lo mismo que la convolución, pero sin invertir el kernel, ya que a medida que  $m$  aumente, el kernel debe disminuir; el desplazamiento del kernel para una imagen 2D se logra definir de la siguiente manera:

$$S(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Dentro de las CNN, se emplean varias operaciones especiales que afectan la dimensionalidad de los mapas de características obtenidos en cada capa convolucional. Estas operaciones incluyen el relleno (padding), que consiste en añadir píxeles alrededor de la imagen para que el kernel pueda aplicarse completamente sin perder información en los bordes. Luego, está la zancada (stride), que determina la distancia entre cada paso del kernel al desplazarse sobre la imagen. Finalmente, se utiliza el agrupamiento máximo (Maxpooling), que también depende de la zancada. El Maxpooling ayuda a reducir la dimensionalidad de los mapas de características al aplicar un kernel de tamaño 2x2 y una zancada de 2, lo cual reduce a la mitad las dimensiones de las salidas de las capas convolucionales configuradas de esta manera. Este proceso es fundamental para extraer características relevantes y gestionar eficientemente la información en redes neuronales convolucionales, especialmente en tareas como el procesamiento de imágenes y la visión por computadora.

**YOLO (You Only Look Once).** Es un enfoque innovador en el campo del reconocimiento de objetos, detección, clasificación y segmentación de imágenes que se destaca por su velocidad y precisión. A diferencia de los métodos tradicionales que dividen estas tareas en múltiples módulos separados, YOLO consolida todos los algoritmos en un único módulo integrado. Utiliza

arquitecturas de CNN para extraer características clave de las imágenes [17]. Este algoritmo es ampliamente reconocido en el ámbito del Deep Learning debido a sus significativas ventajas en el procesamiento de imágenes, que incluyen eficiencia en tiempo real, robustez y precisión. Ha evolucionado a lo largo de diversas versiones, para el desarrollo de este proyecto se consideró la v8.

**YOLOv8.** Este es un algoritmo diseñado para detectar y clasificar objetos en imágenes con una notable velocidad y precisión. Este algoritmo se basa en el éxito de sus predecesores, abandonando sus limitaciones e incorporando técnicas avanzadas para mejorar su rendimiento. Este adopta una estrategia de entrenamiento integral para optimizar su rendimiento. Una característica notable es el uso de múltiples resoluciones de entrenamiento, permitiendo al modelo aprender de imágenes a diferentes escalas. Además, utiliza la técnica de aumento de datos llamada mosaico (consiste en combinar múltiples imágenes en una sola entrada de entrenamiento), mejorando la capacidad del modelo para generalizar en diversos escenarios.

YOLOv8 trae distintas variantes adaptadas a casos de uso específicos (YOLOv8-C, YOLOv8-D y YOLOv8-E), representas diferentes tamaños del modelo, siendo YOLOv8-D la configuración predeterminada del algoritmo. Ha demostrado ser notable en conjuntos de datos como COCO y VOC, su precisión, combinada con capacidad de procesamiento en tiempo real, lo convierte en una de las mejores opciones.

La base de este algoritmo, conocida como Backbone Network, es responsable de la extracción de características de la imagen de entrada utilizando CSPDarknet53. Esta arquitectura introduce una conexión Cross-Stage Partial (CSP), que mejora el flujo de información entre las diferentes etapas de la red y optimiza el flujo de gradientes. Otra pieza esencial de la arquitectura de este algoritmo es el uso de la Path Aggregation Network (PANet), facilita el flujo de información a través de diferentes resoluciones espaciales, permitiendo al modelo capturar características de múltiples escalas. Esta estructura trabaja en conjunto con múltiples "cabezas" de detección, cada una encargada de predecir cajas delimitadoras, probabilidades de clase y puntuaciones de presencia de objetos. Finalmente, el algoritmo incorpora una función de asignación dinámica de anclaje y una función de pérdida IoU (Intersection over Union). Estas mejoras permiten realizar predicciones más precisas de las cajas delimitadoras y un mejor manejo de objetos superpuestos, conocida como Detection Head.

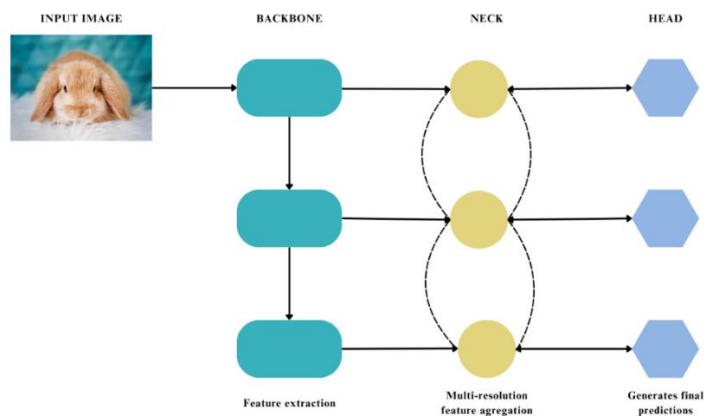


Fig 10. Arquitectura de YOLOv8. (Fuente: Autor)

## 7.2 Estado del Arte.

Para el desarrollo de este proyecto, se llevó a cabo una investigación en las bases de datos institucionales con el objetivo de identificar las técnicas más utilizadas en la clasificación y segmentación de frutos. En los artículos revisados para el desarrollo e implementación del algoritmo de selección de fresas, se descubrió que las técnicas predominantes están directamente asociadas al campo de la inteligencia artificial, salvo algunos casos aislados en los que se emplearon otras técnicas.

Para la revisión de los documentos, se elaboró un cuadro descriptivo que recopila y compara las técnicas que utilizan inteligencia artificial en sus diversas áreas y aquellas que no, específicamente enfocadas en la clasificación y segmentación de frutos. Esta comparación detallada permitió identificar las metodologías más relevantes para la aplicación propuesta en este proyecto.

Tabla 2. Técnicas de Inteligencia Artificial para clasificación y segmentación de imágenes. (Fuente: Autor)

<b>Investigaciones relacionadas</b>	
<b>Descripción</b>	Un método preeminente y de vanguardia para la clasificación de imágenes es la utilización de CNN, gracias a su capacidad automática de detección de patrones y su rendimiento probado como se mencionó anteriormente. En particular, las CNN han demostrado su eficiencia en la clasificación de variedades de fresa en diversas etapas de maduración, logrando niveles de precisión que oscilan entre el 93% y el 97% [18]. Es relevante destacar que las redes neuronales convolucionales, incluyendo arquitecturas como VGG16, ResNet, AlexNet y PANet, han sido entrenadas con conjuntos masivos de imágenes con el objetivo de optimizar su eficiencia.
<b>Resultados</b>	En la investigación, demostraron que las combinaciones apiladas entre los modelos EfficientNet y CNN tiene la precisión al calificar los conjuntos de pruebas y las muestras reales en comparación con otros modelos de aprendizaje profundo.
<b>Descripción</b>	Problemas como la clasificación manual en frutos que son susceptibles a colisiones y plagas durante el crecimiento, recolección, almacenamiento y transporte, son fuentes de grandes investigaciones, para mejorar el producto [19], para esto se adoptaron métodos de aprendizaje profundo basados en una red neuronal convolucional y tecnología de visión por computadora para lograr una clasificación eficiente en cinco categorías. Haciendo uso de técnicas como el optimizador de promedio de peso estocástico (SWA) y la función de pérdida w-softmax para mejorar la red VGG. Dando precisiones oscilantes entre 93.8% al 95.65% con grandes velocidades de procesamiento.
<b>Resultados</b>	Debido al objetivo de la investigación, los resultados se presentan conceptualmente verificando la detección de cambios morfológicos en la imagen del fruto y corroborándolo con matrices de confusión.
<b>Descripción</b>	La extracción de características del color de las imágenes es otra opción de clasificar imágenes, en esta ocasión se utilizó la técnica de extracción

	del histograma a color en tres dimensiones, que de la mano de algoritmos de inteligencia artificial se efectuó la clasificación automática [20].
<b>Resultados</b>	El autor, evaluó distintos tipos de clasificadores bajo diferentes tamaños de muestras, obteniendo una eficiencia superior con el clasificador “Random Forest”, de igual manera optimizo la clasificación basada en la extracción de características del histograma “3D-RGB”.
<b>Descripción</b>	Técnicas innovadoras en la visión artificial aplicadas a la detección de características de fresas para su clasificación en la industria alimentaria es el claro ejemplo del alcance de la inteligencia artificial [21], haciendo usos de redes neuronales las imágenes se procesan mediante técnicas avanzadas de detección de bordes y una región de interés (ROI) dinámica, tras su procesamiento, el algoritmo es capaz de reconocer características importantes como la forma, color y las anomalías en las fresas.
<b>Resultados</b>	El autor presenta los resultados en una tabla, donde demuestra la eficacia del clasificador de un 90% a 93%. De igual manera, demostraron que la inclusión de varios Agentes Neuronales Entrenables Aisladamente (ANEA’s) crearon fuertes enlaces entre las imágenes y los enlaces lo cual hace posible la clasificación del fruto bajo condiciones de luz variable.
<b>Descripción</b>	El uso de modelos como Mobile Net de TensorFlow, son de gran ayuda al momento de clasificar imágenes, ya que con su gran procesamiento y capacidad de entrenamiento son eficaces en estos campos. Esta investigación [22], se basó en la clasificación de 13 categorías de frutos, las imágenes tomadas fueron sometidas a proceso de etiquetamiento y segmentación por su calidad de resolución, mejorando la eficiencia del entrenamiento.
<b>Resultados</b>	La validación de los resultados la realizaron usando los tiempos de respuesta y aciertos que obtuvieron en la predicción del modelo de red neuronal, logrando obtener un 87% de eficiencia.
<b>Descripción</b>	La calidad del producto es uno de los factores más importantes al momento de querer exportar un producto, por ello esta investigación, se basó en la unión de la espectroscopia (descomposición de la luz y mide diferentes longitudes de onda de luz visible y no visible) con técnicas de Machine Learning y Deep Learning para la detección y clasificación de arándanos [23], De igual manera, permitió reducir los costos, el tiempo y optimizo todo el proceso industrial. Esto se logró digitalizando los espectros NIR de las muestras de arándanos y entrenándolos en una red neuronal, obteniendo una exactitud del 98% en la clasificación.
<b>Resultados</b>	La investigación, hizo uso de la métrica de matriz de confusión para obtener su rendimiento, demostrando una buena eficacia del 92%.
<b>Descripción</b>	Existen diferentes maneras de realizar una clasificación, sin embargo, esta investigación apporto [24] un peso en el desarrollo del documento, este se enfocó en el procesamiento de las imágenes para la identificación de frutos haciendo uso de lenguajes de código abierto como OpenCv y TensorFlow, lograron identificar cada uno de ellos, y combinado de robótica, alcanzaron el desarrollo de un clasificador eficiente.

<b>Resultados</b>	La investigación no plasmo resultados en gráficas y no muestran una valoración numérica, sin embargo, demostraron su eficiencia implementándolo ya que era un reto combinar conocimientos de mecánica, electrónica y programación para realizar todo el proceso.
<b>Descripción</b>	Un estudio reciente [25], implemento la clasificación usando múltiples técnicas de Visión por Computador como: detección de bordes de Canny, operaciones morfológicas, segmentación de fondo por umbrales, entre otras técnicas que permiten la extracción de características, permitiendo dar paso a una estructura de clasificación condicional. Junto con estas técnicas los autores realizaron pruebas con técnicas de Machine Learning para finalmente implementar CNN, que ajusto específicamente con parámetros del aprendizaje de transferencia.
<b>Resultados</b>	Los autores, demostraron la viabilidad de la investigación realizada por medio de métricas y plasmando en graficas que no requiere de un gran dispositivo de cómputo para su implementación, de igual manera compararon todas las técnicas que utilizaron en el transcurso de este, demostrando las ventajas y desventajas que les proporciono cada una de ellas.
<b>Descripción</b>	La madurez del fruto es otro de los factores importantes al momento de vender o comprar un producto, generalmente se requiere de un experto que lo analice. Un estudio [26], realizo un clasificador según el estado de madurez del fruto; para determinarlo se llevó cada imagen a una transformación de 4 espacios de color (RGB, CIElab, YCbCr y HSV) obteniendo 12 características, para ser sometidas a tres clasificadores los cuales determinaron su estado de madurez.
<b>Resultados</b>	La investigación, plasmo los resultados estadísticos de error en tablas comparando entre la precisión de etiquetas “inmadura” y “en maduración”, observando tener respuestas de hasta el 90% de exactitud.
<b>Descripción</b>	En estudios recientes, se implementó YOLOv8 en conjuntos con fresas para la tarea de detección. En este enfoque, se utilizaron redes residuales con parámetros aprendibles y normalizados para mejorar la eficiencia de la red de fusión de características del algoritmo específico, LS YOLOv8s. La validación de 5 veces evidencio la superioridad de este modelo sobre otros algoritmos en comparación con YOLOv5, CenterNet y SSD [27].
<b>Resultados</b>	El autor valido el algoritmo demostrando mejoras del 1.6%, 33.5% y 3.4% en mAP0.5 en comparación con los algoritmos mencionados anteriormente. Además, demostró que LS-YOLOv8s supera a YOLOv8m en precisión y velocidad de detección, utilizando menos parámetros. Estos resultados respaldan la confiabilidad de LS-YOLOv8s y refuerzan la posición de YOLOv8 como uno de los principales algoritmos para aplicaciones de inteligencia artificial en la industria.
<b>Descripción</b>	Existe una gran variedad de tipos de frutas, cada una con propiedades físicas similares y ligeras diferencias en color, forma y carnosidad. Debido a la gran variedad de características se convierte en un desafío clasificar todos los tipos de frutas. El propósito de [28] fue utilizar la tecnología de aprendizaje profundo para detectar y clasificar usando los algoritmos de

	YOLO, entrenando con diferentes modelos (YOLOv5, YOLOv7, YOLOv8), utilizando 1735 imágenes con 9 tipos de frutas, los entrenamientos fueron realizados en 100 <i>épocas</i> junto con <i>batch</i> de 16.
<b>Resultados</b>	Con base a los resultados entregados en la investigación, se muestra una dominancia de YOLOv8 logrando un recall medio de 0.99% y una precisión de 0.991% con una IoU coloreada en el rango de [0 – 1] sobre los demás modelos.
<b>Descripción</b>	La tarea de identificación de frutas implica el discernimiento y la clasificación de varias variedades de frutas, en función de sus atributos visuales. Este acto se logra a través de varios métodos tanto manuales como de técnicas tradicionales de visión por computadora. Este trabajo [29] logró reconocer 15 tipos de frutas haciendo uso de YOLOv8, debido a sus Redes Neuronales Convolucionales (CNN) predicen simultáneamente múltiples cuadros delimitadores y clases de probabilidad a partir de estos cuadros. YOLO es uno de los modelos mas populares en el campo de la detección y clasificación de objetos, el entrenamiento de la imagen y la optimización directa del rendimiento de clasificación son sus mayores cualidades.
<b>Resultados</b>	El autor desarrollo esta investigación haciendo uso del modelo Yolov8n, entregando resultados con una precisión promedio de 0.97% durante un proceso de aprendizaje automático de 100 épocas.
<b>Descripción</b>	La clasificación por mano humana ha sido tradicionalmente responsable de determinar la madurez de los frutos. Esta investigación [30] se centra en la detección automatizada de la madurez de los racimos de frutos frescos de palma. La visión por computadora y la inteligencia artificial son cada vez más efectivas para evaluar madurez, sin embargo, muchos métodos requieren operaciones complejas, configuraciones controladas o calibraciones manuales. Este estudio hace hincapié en investigar el potencial del algoritmo YOLOv8, en particular la variante YOLOv8m para la clasificación y detección de la madurez.
<b>Resultados</b>	La investigación arrojo un mAP50-95 de 0.927, demostrando que este modelo equilibra la eficiencia computacional y la precisión, indicando un gran potencial para revolucionar los procedimientos de evaluación de la fruta en la industria.
<b>Descripción</b>	El desarrollo de la inteligencia artificial ha dado lugar a avances notables en la agricultura. Se han desarrollado herramientas automáticas para reducir el trabajo humano y mejorar la precisión. Una de las aplicaciones populares en la cosecha es la clasificación de los frutos basado en su nivel de madurez. Este artículo [31] se centra en mejorar la arquitectura de YOLOv8n mediante la sustitución de las operaciones convolucionales originales por nuevo método módulo de convolución denominado “Modulo de atención de bloques de convolución de campo receptivo para la detección de la madurez de la fruta” para mejorar la extracción de características.
	Los experimentos se evaluaron sobre la maduración de 2 tipos de frutas (mango y plátano). La red propuesta logró un mejor rendimiento con un

	99.4% de mAP-0.5, demostrando una superioridad sobre los otros métodos en las mismas condiciones, sin embargo, la arquitectura original de YOLO no se queda atrás con una diferencia de 0.1%.
<b>Descripción</b>	Esta investigación tuvo como objetivo mejorar un algoritmo para la detección de la madurez de las fresas [32]. El autor propuso un método de reconocimiento preciso de la madurez combinando el aprendizaje profundo y el procesamiento de imágenes. El desarrollo se estructuró en tres etapas. En la primera etapa, se añadieron convoluciones autocalibradas a la red troncal de Mask R-CNN para mejorar el rendimiento del modelo. En la segunda etapa, se utilizaron técnicas de segmentación para extraer las características de color de los canales B, G, L, a y S. Finalmente, en la tercera etapa, se clasificó la madurez de las fresas según los valores de las características de color y se visualizaron los resultados obtenidos.
<b>Resultados</b>	El autor señaló que el rendimiento del algoritmo mejoró sustancialmente con la incorporación de convoluciones autocalibradas en Mask R-CNN, aumentando la robustez contra diversas interferencias de oclusión y llevando su precisión final a 0.937. Esto representa un incremento de 0.039 en comparación con el algoritmo original. En conclusión, se destaca que los resultados de clasificación obtenidos son superiores a los métodos comunes de extracción de características manuales, como lo son los modelos AlexNet y ResNet18.
<b>Descripción</b>	Aunque esta investigación [33] no se centra en la clasificación o determinación de la madurez de un fruto específico, utiliza la técnica de segmentación por umbrales para realizar el conteo de pitayas en el árbol, facilitando su recolección. El autor propuso un nuevo algoritmo llamado Otsu, que se basa en la segmentación de imágenes, la determinación de límites, el análisis de forma y el análisis de superposición para lograr un conteo preciso.
<b>Resultados</b>	El autor no presenta una métrica de evaluación cuantitativa.
<b>Descripción</b>	Aunque la visión artificial se orienta principalmente a la extracción de características, también puede aplicarse eficazmente en la clasificación. Utilizando un sistema de visión basado en redes neuronales con una estructura similar a una DNN, entrenado con aprendizaje robusto frente al ruido, se logra una fuerte relación red-objeto. Esta combinación permite el reconocimiento de características complejas de manera efectiva [34].
<b>Resultados</b>	El autor presenta los resultados en una tabla de métricas, demostrando resultados entre 90% y el 100% de reconocimiento de patrones en las fresas.
<b>Descripción</b>	La ecualización de histograma se utiliza para mejorar el contraste en una variedad de aplicaciones debido a su función simple y efectiva, sin embargo, se puede encontrar el inconveniente de que el brillo de una imagen cambie luego de la ecualización del histograma. Esta investigación [35] propone una nueva extensión de la ecualización de histograma para superar dicho inconveniente. El algoritmo propuesto, utiliza ecualizaciones independientes sobre dos subimágenes obtenidas

	descomponiendo la imagen de entrada en función de su media con la restricción de que las subimágenes resultan limitadas entre sí alrededor de la media de entrada.
<b>Resultados</b>	Se demuestra matemáticamente que el algoritmo preserva significativamente el brillo medio de una imagen, superando a la equalización de histograma tradicional, mientras mejora el contraste.
<b>Descripción</b>	El desarrollo de sistemas para evaluar la calidad de un producto es indispensable para determinar su valor agregado, esta investigación [36] se centra en determinar la calidad del café en el preprocesamiento de imágenes digitales utilizando algoritmos basados en la teoría de Retinex llamado Retinex multiescala con restauración de color (MSRCR). Las imágenes de granos de café se someten a una técnica de matriz de coocurrencia de escala de grises para la extracción de características, junto con una máquina de vectores de soporte (SVM) para evaluar los resultados.
<b>Resultados</b>	La investigación mostró que la técnica MSRCR proporcionó resultados de alta calidad visual y una mayor precisión en la clasificación en comparación con otras técnicas. Específicamente, se alcanzó un nivel de exactitud del 79.1538% y un tiempo de ejecución reducido de 3.280 ms.
<b>Descripción</b>	En la actualidad, existe una amplia variedad de algoritmos y técnicas para abordar problemas relacionados con imágenes digitales. Sin embargo, para enfrentar desafíos específicos como la reflectividad desigual de la luz en superficies, raramente se recurre a técnicas avanzadas de mejora de imágenes digitales, como se observa en [37]. En la detección de defectos basada en segmentación de imágenes, se logró eliminar la interferencia causada por la inhomogeneidad en la reflexión de la luz mediante la corrección de luminancia con Retinex de Escala Única (SSR), que emplea un filtrado bilateral adaptativo. Adicionalmente, se aplicaron correcciones de gamma para optimizar la capa reflectante, la cual fue segmentada posteriormente utilizando un algoritmo de crecimiento de regiones.
<b>Resultados</b>	Los datos estadísticos presentados en la investigación son altamente efectivos, mostrando una precisión del 94.2% en la detección de cicatrices. Estos resultados evidencian que las técnicas empleadas son capaces de detectar de manera eficiente la presencia de una reflectividad de luz desigual.

## 8. Desarrollo del proyecto

Este proyecto utilizó una metodología de investigación cuantitativa de tipo experimental, ya que se requirió la adquisición de datos validados mediante simulaciones y dispositivos físicos. En el desarrollo de este, se implementaron las técnicas de inteligencia artificial estudiadas en la sección anterior. (7).

Antes de iniciar con el núcleo de esta investigación, se debe tener en cuenta que el marco de referencia para el autor está fundamentado en el conocimiento empírico de los agricultores de la región de Boyacá; quienes conocen en detalles el aspecto visual de las categorías, que normalmente se comercializan.

En el desarrollo de la investigación, se tienen en cuenta 5 fases principales, ilustradas en la (Fig 11) como el proceso realizado para obtener una respuesta del algoritmo en cuanto a su rendimiento. Su composición radica en las fases del proyecto, donde A) Adquisición y Manejo de Muestras, B) Clasificación de las muestras aplicando YOLOv8, C) Análisis de datos según la clasificación, D) Determinación del grado de madurez. Estas fases estarán descritas alrededor de esta sección y podrá ser de utilidad para la comprensión de la metodología utilizada en el proyecto.

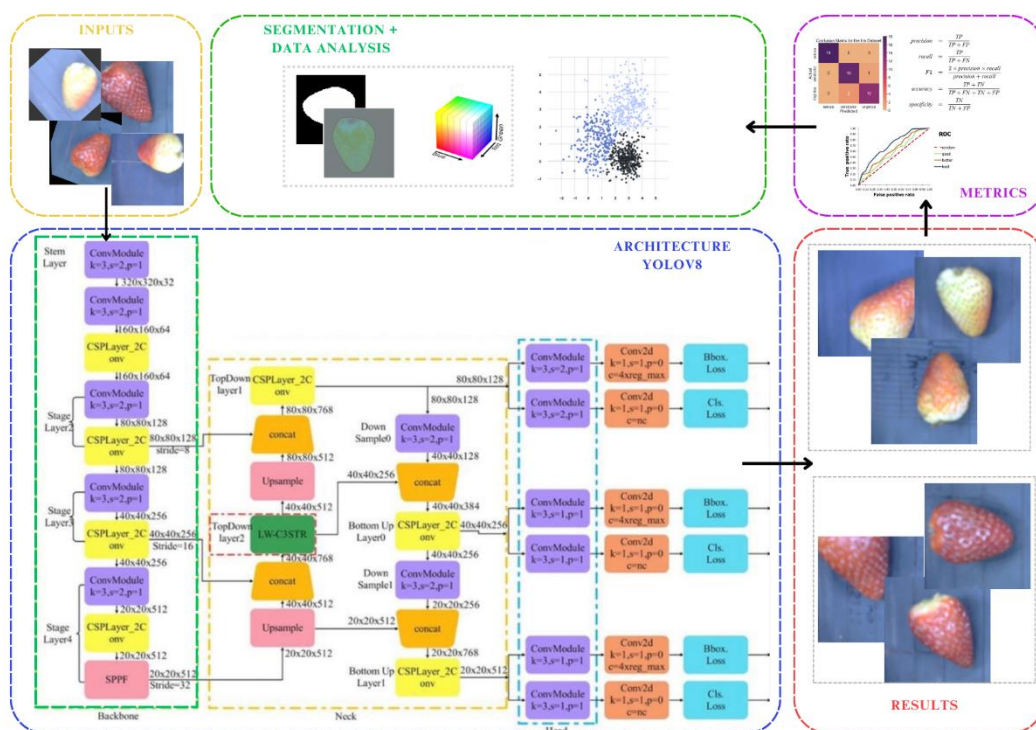


Fig 11. Arquitectura de trabajo llevado a cabo para la clasificación y determinación de madurez de fresa. (Fuente: Autor)

### 8.1 Adquisición y manejo de muestras.

La creación de un dataset es un proceso fundamental para el entrenamiento y evaluación de modelos de Deep Learning. Este es un proceso meticuloso que requiere tiempo y dedicación para generar conjuntos de datos suficientemente grandes, asegurando así un entrenamiento efectivo del modelo y ciertas garantías de rendimiento. Aunque existen numerosos datasets

disponibles en diversas plataformas, muchos de los cuales ya están etiquetados, uno de los objetivos de este proyecto es empezar desde cero, es decir, crear un dataset propio con datos recolectados y etiquetados específicamente para este fin. Para la adquisición de muestras, se utilizó una banda transportadora que fue construida y empleada en una investigación previa [25].

Durante el proceso de recolección de muestras de fresas, se descartaron aquellas con irregularidades relacionadas con la sobre maduración, así como imágenes de fresas cortadas o aquellas que presentaban cualquier otra imperfección que pudiera afectar su representación precisa. Las imágenes que superaron la fase de procesamiento de datos fueron agrupadas según sus etiquetas y posteriormente en tres subconjuntos:

El total de muestras extraídas originalmente fueron 1133, de las cuales 541 corresponden a la primera categoría y 480 a la segunda categoría. Las 112 muestras restantes se apartaron en un lote donde se encuentran muestras de ambas categorías, que fueron sometidas al algoritmo elaborado, donde las predicciones se corroboraron con un experto en el tema. En la Tabla 3, se muestra más descriptivamente el conjunto de muestras tomado.

Tabla 3. Conjunto de muestras tomadas, para el entrenamiento de YOLOv8. En la parte inferior se pueden observar muestras en conjunto de datos. (Fuente: Autor)

<b>Categoría</b>	<b>Número de Muestras</b>		
HalfRipe	541		
Ripe	480		
Prueba	112		
<b>Total</b>	<b>1133</b>		

Las muestras visualizadas en la Tabla 3, se encuentran distribuidas en carpetas que poseen la etiqueta de categoría. Esta carpeta es ingresada al ambiente creado en Python, el cual encontró 1133 imágenes separadas en tres subconjuntos, entrenamiento (70%), validación (20%) y prueba (10%).

Es importante aclarar que el entrenamiento de la red neuronal basada en la arquitectura de YOLOv8 se realizó utilizando el procesador de un ordenador convencional, empleando la

biblioteca Ultralytics. Posteriormente, el modelo se probó en la banda transportadora previamente mencionada. Las razones detrás de esta elección se discutirán en detalle en la sección de discusión.

Se realizó una primera prueba de clasificación, cuyos resultados mostraron un buen rendimiento sobre las imágenes de prueba. Sin embargo, para asegurar la robustez y la capacidad de generalización del modelo de clasificación de fresas desarrollado en este proyecto, se empleó una técnica específica que se detalla en el siguiente apartado (8.1.1). Esta estrategia benefició el aprendizaje del modelo, permitiendo identificar y clasificar fresas con mayor precisión y consistencia.

### 8.1.1 Aumento de datos.

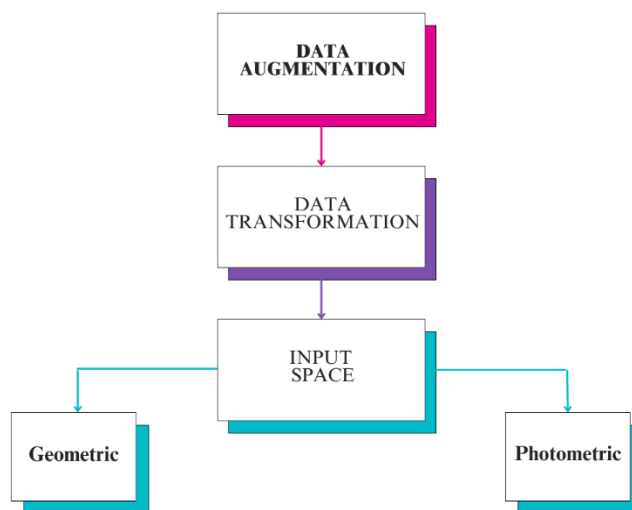


Fig 12. Taxonomía de los enfoques de aumentos de datos en esta investigación. (Fuente: Autor)

Esta técnica es crucial para evitar el sobreajuste (overfitting), un problema común en el entrenamiento de redes neuronales donde el modelo se ajusta demasiado bien a los datos de entrenamiento, pero falla al generalizar datos nuevos y no vistos [38]. El aumento de datos amplió el conjunto de entrenamiento mediante transformaciones aplicadas a las imágenes existentes y si fuera necesario a sus respectivos cuadros delimitadores. Posteriormente, estas imágenes transformadas se añadieron al dataset junto con las originales, esto permitió incrementar la cantidad de datos etiquetados para el entrenamiento de la red. Además, las transformaciones sobre las imágenes generaron nuevos ejemplos, ampliando la variedad de patrones y mejorando la capacidad de generalización de la red durante el proceso de aprendizaje.

### 8.1.2 Transformaciones.

En este apartado, se implementan diversos acondicionamientos a las imágenes que serán procesadas y clasificadas (ver Fig 11). Se emplean métodos como recorte aleatorio, volteo horizontal, rotación aleatoria, ajuste de brillo y transformaciones de tono para modificar la apariencia y el color de las imágenes. Estas transformaciones permiten crear nuevas imágenes realistas y representativas de las variaciones que el modelo puede encontrar en el mundo real, mejorando así la capacidad de generalización del modelo.

El aumento de datos en el proyecto se implementó utilizando la herramienta RoboFlow. Esta herramienta ofrece la creación de conjuntos de datos colaborativos y la asignación de roles de etiquetador a diferentes personas, lo que permite que cada participante aporte información al conjunto de datos. En este caso, se contó con la colaboración de otra persona que ayudó en la tarea de etiquetado. Además, RoboFlow facilita el despliegue en diferentes entornos, como YOLO, que se utilizó en el presente proyecto.

En la Fig 13 se puede observar un ejemplo de etiquetado de una imagen utilizada en el dataset. En la parte izquierda se observa la clase a la que pertenece la imagen en cuestión, teniendo en cuenta, que las imágenes se pueden etiquetar entre las clases ‘HalfRipe’ y ‘Ripe’.

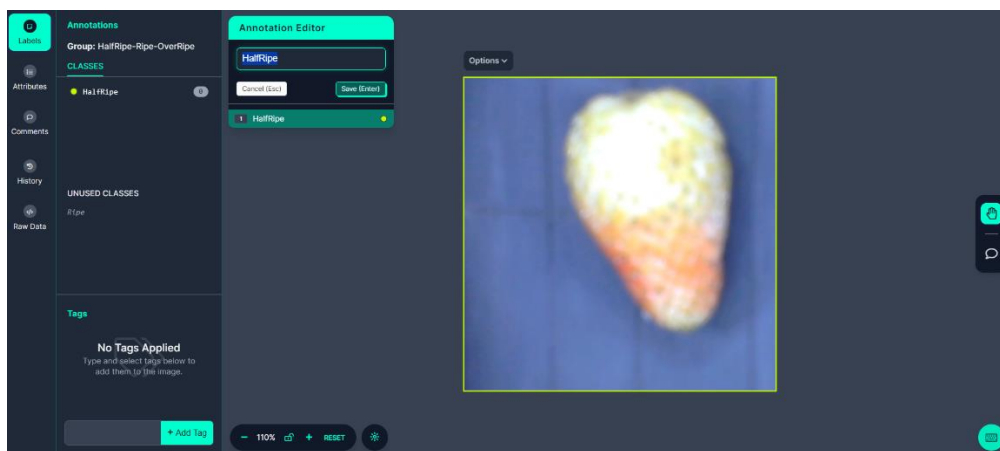


Fig 13. Interfaz de etiquetado de Roboflow. (Fuente: Autor)

Una vez etiquetadas todas las imágenes se continúa a la etapa de procesado. En cuanto al procesado, se aplica un cambio de tamaño en las imágenes para que sean de 640x640 y ese estandariza el orden de los píxeles utilizando auto orientación.

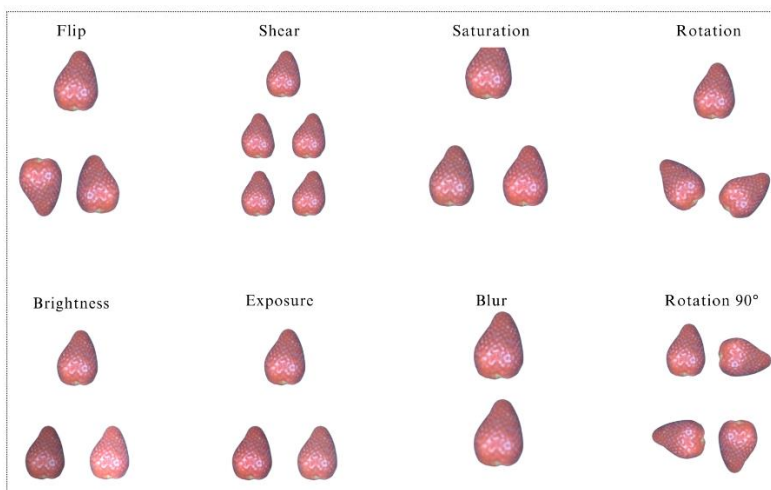


Fig 14. Modelos utilizados para el aumento de datos del dataset original. (Fuente: Autor)

Realizado el procesamiento de las imágenes, se genera el aumento de datos aplicando los modelos que se estipulan convenientes al dataset que se posee (ver Fig 14) cabe resaltar que la generación de datos aumentados ocurre independientemente del entrenamiento. Esto debido que

el tiempo de entrenamiento se acelera porque no gasta recursos informáticos esperando que la CPU cree imágenes aumentadas durante el tiempo de entrenamiento. El aumento de datos incluyó un total de ocho técnicas diferentes: volteo (horizontal y vertical), rotación de 90° (en sentido horario, antihorario, arriba y abajo), rotación (en un rango de -45° a 45°), recorte ( $\pm 5^\circ$  horizontal y  $\pm 8^\circ$  vertical), ajustes de saturación (entre -10% y 10%), ajustes de brillo (entre -20% y 20%), ajustes de exposición (entre -3% y 3%) y, finalmente, difuminado con un radio superior a 2.5 píxeles. En la Tabla 4 se logra observar el dataset distribuido una vez finalizado el aumento de datos para el entrenamiento a realizar en la red.

Tabla 4. Construcción y distribución del dataset obtenido. (Fuente Autor)

<b>Categoría</b>	<b>Muestras de entrenamiento</b>	<b>Muestras de validación</b>	<b>Muestras de prueba</b>
<b>Half Ripe</b>	1236	120	118
<b>Ripe</b>	996	107	95
<b>Subtotal</b>	2232	227	213
<b>Total</b>	<b>2672</b>		

## 8.2 Clasificación de las muestras aplicando YOLOv8.

Para llevar a cabo la tarea de entrenamiento, se requiere un modelo capaz de ejecutar un gran número de operaciones de manera paralela. En cuanto a la implementación del algoritmo para la clasificación de fresas, se propuso el uso del algoritmo preentrenado YOLOv8, este fue seleccionado sobre otras arquitecturas mencionadas en la sección 7.2 debido a la capacidad de clasificación en tiempo real, lo que resulta esencial para aplicaciones donde la velocidad y la eficiencia son cruciales, como en la clasificación de fresas postcosecha. YOLO equilibra la precisión con la velocidad de procesamiento en el manejo de múltiples clases y objetos simultáneamente en bajo tiempo de inferencia, siendo un gran candidato para tareas de clasificación en lotes grandes de imágenes, proporcionando resultados satisfactorios permitiendo el uso de dispositivos con recursos limitados, lo cual es beneficioso para aplicaciones en entornos agrícolas donde la infraestructura tecnológica puede ser básica.

Cabe mencionar que, aunque YOLOv8 es habitualmente entrenado en conjuntos de datos ImageNet para modelos de clasificación de imágenes, se optó por someter las imágenes adquiridas a diversos modelos (Tabla 5) con el objetivo de comparar la eficacia de cada uno sobre el conjunto de datos adquirido, generando así un nuevo algoritmo de clasificación de fresas.

Tabla 5. Modelos de YOLOv8 usados para el entrenamiento. (Fuente: Autor)

<b>Modelo</b>	<b>Tamaño (píxeles)</b>	<b>Velocidad (ms)</b>
<b>YOLOv8n-cls</b>	224	12.9
<b>YOLOv8s-cls</b>	224	23.4
<b>YOLOv8m-cls</b>	224	85.4
<b>YOLOv8l-cls</b>	224	163.0

Subsecuentemente, con base a los resultados obtenidos se identificó el clasificador con un tiempo de procesamiento que maximiza el entrenamiento de la red (véase la sección 9.1). Para este

propósito, se propuso someter el conjunto de datos a una fase inicial de entrenamiento con el fin de observar el comportamiento de las métricas de precisión y pérdida durante esta etapa, asegurándose de que la arquitectura de red seleccionada fuera adecuada.

### 8.2.1 Entrenamiento del modelo

Antes de comenzar, es fundamental asegurarse de que se está utilizando la GPU de la unidad de cómputo. Para esto, se creó un entorno virtual en el editor de código fuente independiente Visual Studio. En dicho entorno, se instaló la versión 8.0.20 de YOLOv8, importada desde Ultralytics, el desarrollador de esta arquitectura.

Tabla 6. Pseudocódigo de Clasificación. (Fuente: Autor)

---

**Algoritmo 1:** Pseudocódigo del Maestro para clasificación usando YOLO

---

- 1 Importar la biblioteca YOLO de Ultralytics
- 2 Cargar el modelo YOLO preentrenado
- 3 Entrenar el modelo en función de los siguientes parámetros:
- 4     | Ruta al conjunto de datos
- 5     | Número de épocas
- 6     | Tamaño de las imágenes
- 7 Guardar los resultados del entrenamiento
- 8 Validar el modelo y guardar los resultados de la validación

---

Con el conjunto de datos previamente organizado, las imágenes son procesadas durante el entrenamiento a través de varias capas convolucionales para extraer características relevantes como bordes, texturas y formas, que son esenciales para la identificación de objetos. YOLOv8 utiliza un enfoque unificado en el que la detección de múltiples objetos y la clasificación se realiza simultáneamente. Las características extraídas pasan por una serie de capas de predicción generando mapas de características, los cuales se comparan con las etiquetas de las clases de los datos de entrenamiento, mientras que la función de pérdida mide la diferencia entre las predicciones del modelo y las etiquetas reales. A través de un proceso iterativo de retropropagación, el modelo ajusta sus pesos internos para minimizar esta función de pérdida, esto ocurre de manera cíclica hasta que el modelo alcanza un nivel satisfactorio en la clasificación de imágenes.

Como se mencionó anteriormente, se utilizaron todos los modelos disponibles para determinar cuál era el adecuado para el conjunto de datos. El entrenamiento de estos modelos se realizó con 10 y 100 épocas. El parámetro *epochs* (épocas) es un hiperparámetro que representa una iteración completa de procesamiento de un conjunto de datos durante el entrenamiento de un modelo.

Durante el entrenamiento del modelo, los datos se dividen en pequeños lotes llamados *batch*. Cada lote se introduce en el modelo, el cual calcula el error en función de la salida del modelo y ajusta sus parámetros para reducir dicho error. Cada ciclo completo a través de todos los lotes de entrenamiento se denomina una época. Es importante tener cuidado con el ajuste de este hiperparámetro, ya que al entrenar el modelo durante más épocas puede permitirle aprender más y reducir el error, pero también puede llevar al sobreajuste.

### 8.3 Análisis de datos según la clasificación.

Es crucial asegurar la calidad de un modelo de aprendizaje profundo mediante una evaluación, que permita discriminar entre diferentes modelos de la arquitectura para facilitar su selección óptima. En esta investigación, se utilizaron diversas métricas para validar el modelo desarrollado, asegurando su eficacia [39].

En Machine Learning, una de las herramientas esenciales para evaluar modelos es la matriz de confusión (confusion matrix). Esta herramienta consiste en una tabla que compara las predicciones del modelo con los valores reales, distribuyendo los resultados en filas y columnas. Utilizamos esta tabla para comprender mejor el desempeño de los modelos y para identificar explícitamente cuando una clase es confundida con otra. En el caso de esta investigación, que emplea un clasificador binario, la matriz de confusión tiene la siguiente estructura (Fig 15):

En la que:

- TP son los positivos que fueron clasificados correctamente como positivos por el modelo.
- TN son los negativos que fueron clasificados correctamente como negativos por el modelo.
- FP son los negativos que fueron clasificados incorrectamente como positivos.
- FN son los negativos que fueron clasificados incorrectamente como positivos.

		ACTUAL VALUES	
		POSITIVE (y)	NEGATIVE (n)
PREDICTED VALUES	POSITIVE (y)	True Positive (TP)	False Positive (FP)
	NEGATIVE (n)	False Negative (FN)	True Negative (TN)

Fig 15. Matriz de confusión. (Fuente: Autor)

Con esta matriz, se calculó la exactitud (*accuracy*), una métrica de evaluación que, aunque comúnmente utilizada, puede ser engañosa respecto a la calidad del modelo. Esto se debe a que la exactitud no diferencia entre errores de tipo FP y FN, tratándolos como si tuvieran la misma importancia. La exactitud se calcula sumando los valores de las diagonales de la matriz de confusión y dividiendo este total por el número total de observaciones en la matriz:

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

Debido a la poca confiabilidad que brinda la exactitud, comúnmente se respalda con otra métrica como la sensibilidad (*recall*), que indica como de bien el modelo evita los FN:

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

Por último, para confirmar la eficiencia de los modelos, se utilizaron la métrica *F1-score* y la curva *ROC*. Estas herramientas, basadas en las métricas previamente obtenidas, revelan diferencias en el rendimiento de un clasificador que no pueden ser identificadas únicamente mediante la *accuracy*.

$$F1\ score = 2 \frac{(accuracy)(recall)}{accuracy + recall}$$

## 8.4 Determinación del grado de madurez

Luego de la clasificación de la fresa, el procesamiento de la imagen se realiza de manera secuencial. Un aspecto crucial es que no se puede simplificar directamente la imagen debido a la presencia del fondo, el cual puede ser confundido por el sistema con las áreas inmaduras y verdosas de la fresa, además de ser sensible a los cambios de iluminación. Para abordar estos desafíos, se propone una combinación de técnicas que aseguren que la imagen tenga las condiciones ideales para la determinación de la madurez de la fresa. Estas técnicas permiten resaltar las características relevantes y mitigar las posibles confusiones, garantizando un análisis más preciso.

### 8.4.1 Preprocesamiento

El preprocesamiento es una etapa crucial en la cual se realizan acondicionamientos a las imágenes que serán posteriormente procesadas y segmentadas (ver Fig 11). El uso de técnicas como la conversión de espacios de color, la reducción de dimensionalidad, la aplicación de filtros y algoritmos de Machine Learning orientados a imágenes reduce el tiempo de preprocesamiento y mejora la detección de elementos que no son fácilmente distinguibles a simple vista. Estas técnicas permiten resaltar características importantes y mejorar la calidad de los datos para los pasos subsiguientes del análisis.



Fig 16. Fresa en condiciones ideales. (Fuente: Autor)

En este apartado, ya se empiezan a hacer ciertos acondicionamientos en los frames que llegan de la clasificación. En primera medida, para resaltar características de la imagen que no son evidentes en otros espacios de color, se realizó un cambio del espacio de color BGR (Blue, Green, Red) a HSV (Hue, Saturation, Value). Esta transformación contribuyó a una separación más intuitiva de las propiedades de color, para mejorar la imagen y facilitar la identificación de características, relacionadas con variaciones en el tono, la saturación y la luminosidad.

Además, para mejorar el procesamiento de la imagen, se realizó un redimensionado de la imagen. Redimensionar la imagen implica reducir la cantidad de datos que deben ser procesados, lo que resulta en un menor tiempo de cómputo y en una menor carga de memoria. Este paso es esencial para mejorar grandes conjuntos de datos de manera eficiente, manteniendo la proporción original para evitar distorsiones que puedan afectar la precisión del análisis.

Acondicionada la imagen, se empleó el algoritmo K-means para realizar la segmentación inicial de la fresa. Al aplicar este algoritmo, se logra obtener una máscara preliminar de la fresa, separando de forma efectiva los píxeles correspondientes a la fruta de aquellos que pertenecen al fondo y otros objetos en la imagen.

Con la máscara inicial generada por K-means, se procedió a aplicar operaciones morfológicas para refinarla. Estas operaciones, como la dilatación, la erosión y la apertura, ayudan a eliminar pequeños desperfectos y ruidos en la máscara, permitiendo obtener una representación más precisa y continua de la fresa en la imagen, facilitando la posterior segmentación. Los elementos usados para ejecutar los pasos de preprocesamiento y segmentación fueron: AMD Ryzen 5 3500U, una GPU: AMD Radeon™ Vega 8 y una memoria RAM de 12 GB.



Fig 17. Mascara antes y después de realizadas las operaciones morfológicas. (Fuente: Autor)

Tabla 7. Pseudocódigo de Determinación de Madurez (Primera Parte). (Fuente: Autor)

---

**Algoritmo 2:** Pseudocódigo del Maestro para la determinación de madurez (Primera Parte).

---

- 1 Guardar imagen que llega de la clasificación
- 2 Obtener las dimensiones de la imagen
- 3 Calcular y retornar las nuevas dimensiones con el factor de redimensionamiento
- 4 Redimensionar la imagen
- 5 Conversión de la imagen del espacio de color BGR a HSV
- 6 Inicializar el algoritmo K-means
- 7 Obtener los centros de los clústers
- 8 Conteo de los píxeles de cada clúster
- 9 Crear una máscara inicial en base a los valores HSV específicos:
  - 10 Para cada etiqueta
  - 11 Obtener valores H, S, V
  - 12 Marcar los píxeles de la máscara si los valores H están en el rango deseado
- 13 Crear una máscara para los píxeles negros
- 14 Ajustar la máscara con inicial con la máscara negra

- 15 Aplicar operaciones morfológicas
  - 16 Redimensionamiento de la máscara refinada
  - 17 Aplicar la máscara refinada para segmentar la fresa
- 

#### 8.4.2 *Procesamiento de imagen*

La luminosidad es uno de los parámetros más cruciales en cualquier sistema de visión por computadora. Controlar esta variable, que normalmente es fluctuante, mejora significativamente la fidelidad de los datos adquiridos, en este caso, imágenes [8]. Dado que el sistema está diseñado para determinar la madurez de la fresa, y considerando que este fruto suele presentar brillo después de ser cosechado (debido a los procesos de manipulación), dicho brillo puede convertirse en un obstáculo durante el procesamiento de la imagen. Por lo tanto, al reducir estos brillos, se mejora tanto el procesamiento como la respuesta del sistema.

Aunque los algoritmos realizados en la sección anterior mostraban un rendimiento que cumplía con las expectativas de color y tamaño, aun se presentan dentro de la estructura inconvenientes con la luminosidad en la imagen, es por esto que, se añadieron técnicas de visión por computador y ML mencionadas en el marco teórico.

Para que las características que se necesitan queden identificadas, se optó por manejar de nuevo un espacio de color, ya que luego de haber sido preprocesada la imagen seguía presentando problemas de iluminación. Se utilizó el espacio de color LAB en lugar del espacio de color RGB debido a su capacidad para separar la información de luminosidad (L) de la información de color (A y B). En el espacio RGB, los canales de color están interdependientes, lo que dificulta la manipulación individual de la luminosidad sin afectar los colores. En contraste, el espacio LAB permitió ajustar la luminosidad de manera independiente, lo que es crucial para mejorar la visibilidad de los detalles en la segmentación previamente realizada, como los tonos rojos y verdes de la fresa, esenciales para determinar su madurez.

En el desarrollo de las técnicas de visión por computador, presentadas en la sección anterior permitieron identificar, una de las características que se ven más afectadas es el color, por ello se enfocó en corregir todos los efectos desfavorables de la imagen captada, haciendo uso de la técnica Retinex, la cual se utiliza para corregir el brillo y mejorar el contraste de la imagen. Un modelo aceptado sugiere que cualquier imagen dada S es la multiplicación píxel por píxel de dos imágenes, la imagen reflejada R y la imagen iluminada L [40]:

$$S_{(x,y)} = R_{(x,y)} \cdot L_{(x,y)}$$

Esta técnica se basa en la teoría de la percepción del color humano, siendo que el sistema humano es capaz de mantener la constancia del color bajo diversas condiciones de iluminación. Al aplicar Retinex, se mejoró significativamente la calidad de la imagen en términos de luminosidad y contraste, haciendo que las características relevantes resalten y sean fáciles de detectar. Esta mejora es particularmente importante para minimizar los efectos reflejos, que pueden inferir con la correcta detección de madurez de la fresa.

Hasta el momento el objetivo fue perfeccionar la imagen para la detección de madurez de la fresa, una vez realizada la mejora, se procedió a aplicar el algoritmo K-means para realizar una segunda segmentación, siendo esta vez, sobre la máscara realizada anteriormente. La función de K-means es agrupar los píxeles de la imagen en clústers basados en sus similitudes de color permitiendo una discriminación más fina entre las áreas de la fresa que presentan estados de madurez. Este paso permite identificar los píxeles que corresponden a los tonos rojos y verdes.

Después de aplicar el algoritmo de K-means, se cuenta el número de píxeles pertenecientes a cada clúster identificado como rojo o verde. Estos conteos se utilizan para calcular el porcentaje de madurez de la fresa. El porcentaje se determina como la proporción de píxeles rojos en relación con el total de píxeles rojos y verdes. Este enfoque cuantitativo permitió una medida precisa y objetiva del estado de madurez de la fresa, basada en el análisis detallado de los colores representativos en la imagen.

Tabla 8. Pseudocódigo de Determinación de Madurez (Segunda Parte). (Fuente: Autor)

---

**Algoritmo 2:** Pseudocódigo del Maestro para la determinación de madurez (Segunda Parte)

---

```

1  Function Retinex:
2  | Convertir la imagen a espacio de color LAB
3  | Normalizar la componente de luminosidad
4  | Aplicar desenfoco Gaussiano a la componente de luminosidad
5  | Obtener la componente Retinex
6  | Combinar la componente Retinex y las otras componentes de color
7  Return
8  Function DetermineStrawberryMaturity:
9  | Calcular el total de píxeles en la máscara
10 | Calcular el porcentaje de píxeles rojos sobre la suma de píxeles rojos y verdes
11 | if conteo de píxeles rojos >= conteo de píxeles verdes
12 | | Return Ripe
13 | else
14 | | Return HalfRipe
15 Return

```

---

## 9. Análisis de Resultados y Discusión

En esta sección, se presentan los resultados obtenidos durante el desarrollo de esta investigación, abarcando tanto la etapa de clasificación como la de determinación del nivel de madurez de las fresas. Se detallan las métricas de rendimiento utilizadas para evaluar la efectividad de los modelos. Estos resultados ofrecen una visión detallada del desempeño de los algoritmos, facilitando una mejor comprensión de su capacidad para ser implementados en escenarios reales.

### 9.1 Clasificación.

Para la clasificación de fresas, se consideraron los resultados obtenidos durante el entrenamiento del modelo, observando cómo se ajustaba con el tiempo. En la gráfica presentada a continuación, se visualizan las pérdidas del primer modelo a lo largo del entrenamiento, lo que permite identificar el comportamiento del modelo a medida que se optimizan sus parámetros. La línea de pérdida de entrenamiento (train loss) refleja el error del modelo en los datos utilizados para su ajuste, mientras que la línea de pérdida de validación (val loss) muestra el error en el conjunto de datos no utilizados durante el entrenamiento. Esto proporciona una medida de la capacidad de generalización del modelo.

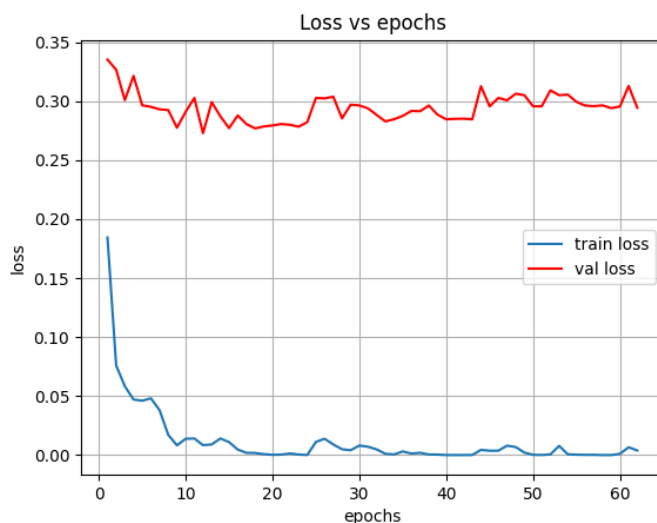


Fig 18. Pérdidas de entrenamiento del primer modelo. (Fuente: Autor)

La gráfica de pérdidas presentada en la Fig 18 ilustra la evolución de la función de pérdida para el conjunto de entrenamiento como para el conjunto de validación a lo largo de 60 épocas. Se puede observar una disminución continua de la pérdida de entrenamiento, lo cual indica que el modelo está aprendiendo de manera efectiva a partir del dataset suministrado. Durante las primeras 10 épocas, la pérdida de validación también disminuyó; sin embargo, a partir de la época 20, se observa un leve incremento en la pérdida de validación mientras que la pérdida de entrenamiento sigue disminuyendo. Este fenómeno es indicativo de sobreajuste, donde el modelo empieza a memorizar los datos de entrenamiento en lugar de aprender patrones generalizables.

Tras identificar el sobreajuste, se implementó la técnica de data augmentation para mitigar este problema y mejorar la capacidad de generalización del modelo, mediante transformaciones como se logra observar en la Fig 14.

Luego del aumento de datos, se reentrenó el modelo utilizando el algoritmo mostrado en la Tabla 6. Los resultados de esta modificación se presentan en la Fig 19, mostrando un aumento en la métrica de precisión y una reducción en la pérdida de validación. Estos efectos contribuyen significativamente a mitigar el sobreajuste en el modelo.

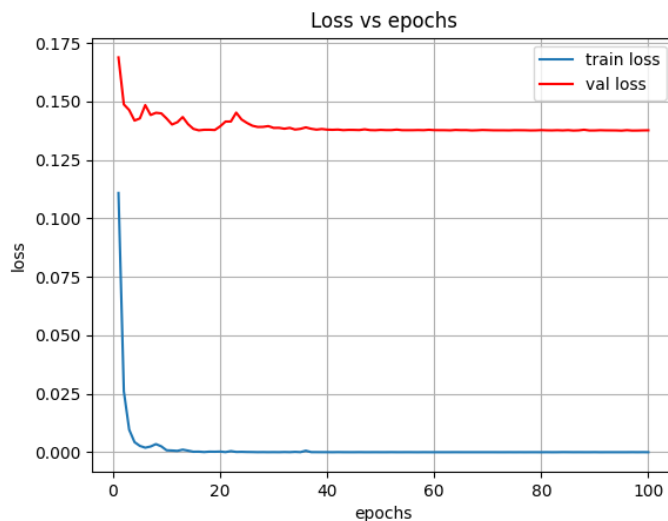


Fig 19. Métrica de entrenamiento del modelo con data augmentation. (Fuente: Autor)

Después de entrenar con el dataset modificado por el aumento de datos, en la Fig 19 se observa que tanto la curva de pérdida de entrenamiento como la de validación disminuyen continuamente hasta estabilizarse sin mostrar cambios en su comportamiento posteriormente. Este patrón sugiere que el modelo está logrando una buena capacidad de generalización a datos no vistos después de la época 20 de entrenamiento.

Los valores de la pérdida de entrenamiento y validación son 0 y 0.14, respectivamente. La diferencia pequeña entre estos valores es un indicativo positivo capacidad del modelo para generalizar.

De igual manera, otra forma para corroborar el desempeño del modelo en Machine Learning son con las curvas de aprendizaje, esta analiza como mejora el rendimiento general del modelo conforme se le proporcionan más imágenes de entrenamiento. La Fig 20, exhibe el rendimiento comparativo entre el primer y segundo modelo.

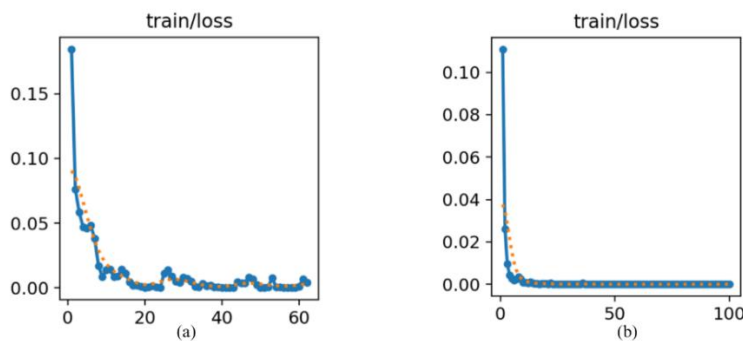


Fig 20. (a) Curva de aprendizaje del primer modelo, (b) Curva de aprendizaje del modelo con data augmentation. (Fuente: Autor)

En la gráfica (a), representa el entrenamiento inicial del modelo de clasificación de fresa sin aumento de datos, donde se observa un patrón característico de sobreajuste. Durante las primeras épocas, las pérdidas de entrenamiento y validación disminuyen de igual manera, sin embargo, a medida que avanzan las épocas no logran estabilizarse en un punto, reafirmando los resultados obtenidos de la Fig 18, sugiriendo un sobreajuste en el modelo.

En contraste, en la gráfica (b) después de aplicar el aumento de datos, se observa un comportamiento significativamente diferente y deseable. La pérdida de entrenamiento muestra una disminución notable, lo que indica que el modelo está aprendiendo eficazmente los patrones presentes en los datos de entrenamiento. En un punto, ambas pérdidas se estabilizan sin mostrar cambios adicionales, lo cual sugiere que el modelo ha alcanzado un buen equilibrio entre precisión y capacidad de generalización. Es destacable que la pérdida de validación también disminuye de manera similar y se mantiene cercana a la pérdida de entrenamiento, lo que indica que el modelo no presenta signos de sobreajuste.

A continuación, se muestran los resultados de las métricas que miden el rendimiento de los modelos presentados hasta el momento. La librería scikit learn permite evaluar varias métricas. A continuación, se muestra la Tabla 9 con las métricas que mejor miden el rendimiento en la tarea de clasificación.

Tabla 9. Evaluación de métricas del modelo antes y después del aumento de datos. (Fuente: Autor)

Métrica	Sin aumento de datos	Con aumento de datos
Accuracy	96%	97%
Precisión	96.3%	97.08%
Recall	96.03%	96.916%
F1- Score	96.02%	96.9%

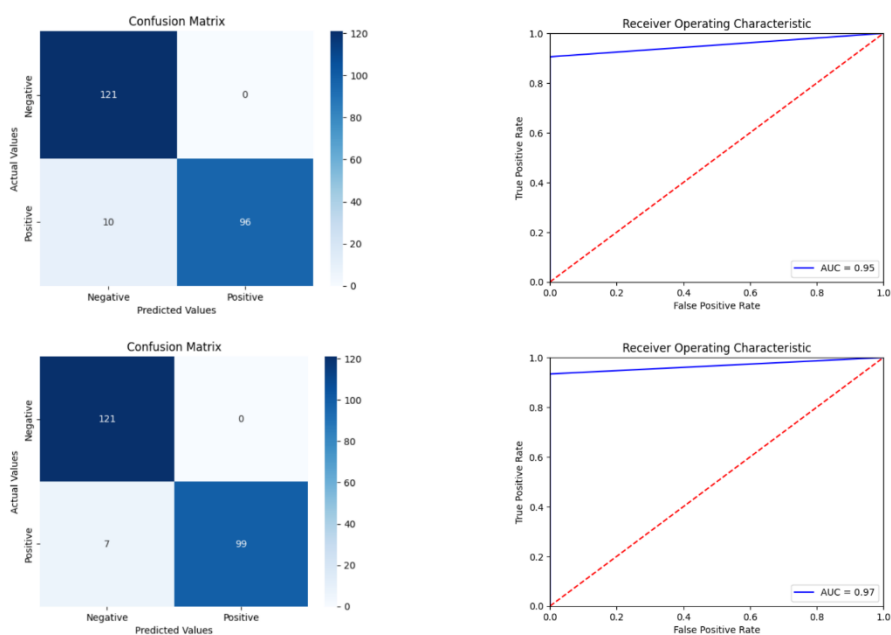


Fig 21. Métricas gráficas de la evaluación de los modelos antes y después del aumento de datos respectivamente. (Fuente: Autor)

## 9.2 Detección de nivel de madurez.

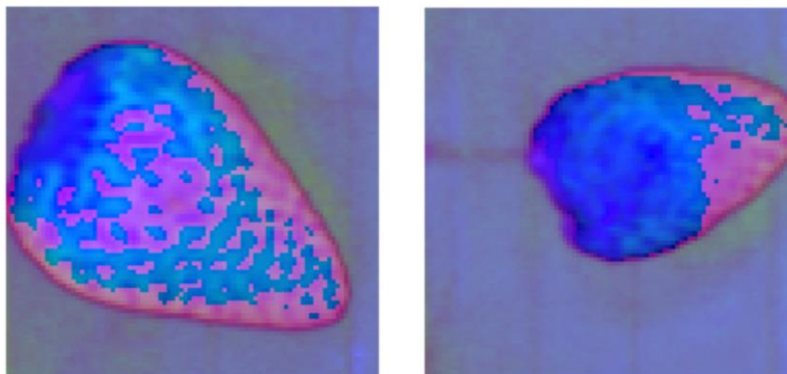


Fig 22. Imagen original en Espacio de Color HSV. (Fuente: Autor)

En la Fig 22, se observa el primer producto obtenido del algoritmo de preprocesamiento, después de la lectura de la imagen se realizó una transformación al color HSV sobre el cual se hace un posterior proceso de simplificación. Es evidente que la caracterización de color en un espacio diferente al RGB presenta una forma más sencilla para mejorar el contraste entre la fresa y el fondo, lo que facilita la segmentación mediante clusterización de colores.

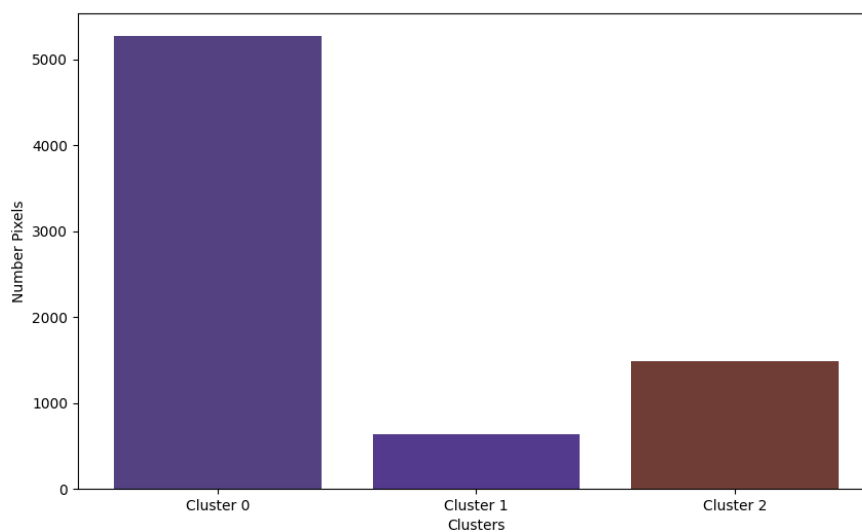


Fig 23. Histograma de clústers en imagen HSV. (Fuente: Autor)

De la imagen HSV de la fresa, se presenta el histograma para identificar cómo se comporta el color en cada uno de los canales de espacio tricromático, se puede observar que la imagen presenta una distribución en el histograma donde las tres componentes están presentes. Al encontrar los umbrales que resaltan el color de la fresa, se aplicó una máscara que logro segmentar el fondo.



Fig 24. Máscara de imagen original. (Fuente: Autor)

Para definir el rango de valores que no serán filtrados en la máscara y los que sí se omitirán, se registraron todos los umbrales de los tonos relacionados con las fresas. Con base en estos resultados, se determinaron los límites superior e inferior del rango de colores que la máscara debe conservar. Además, se aplicaron operaciones morfológicas para corregir las incidencias en los bordes de la máscara.

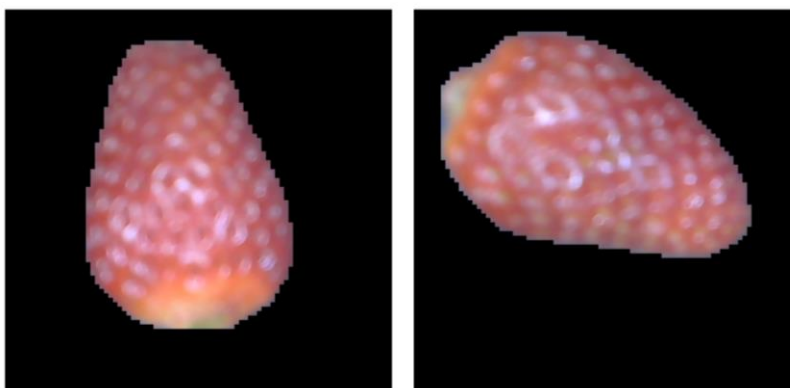


Fig 25. Imagen original segmentada. (Fuente: Autor)

Con los rangos de colores definidos, se indica al algoritmo para que excluya de la imagen todos los píxeles que no se encuentren dentro de esos valores. Se crea una matriz binaria donde los píxeles que cumplen con los criterios se transforman en unos (blancos) y los que no, en ceros (negros). Al aplicar esta máscara a la imagen original mediante una operación de multiplicación, los píxeles que coinciden con los unos en la máscara se mantienen en la imagen resultante, mientras que los píxeles correspondientes a ceros son omitidos.

En la Fig 26 se muestra la transformación de la imagen segmentada a espacio de color LAB, esta transformación logro una mejora significativa en la separación de luminosidad y color, lo que permite una manipulación efectiva de la imagen. En particular, se modificó el canal L utilizando la técnica Retinex, mejorando la percepción de contraste y detalles en la imagen.

Esta transformación facilita una mejor discriminación de las regiones de interés, destacando las características relevantes de la fresa, y proporcionando una base sólida para la posterior clusterización.

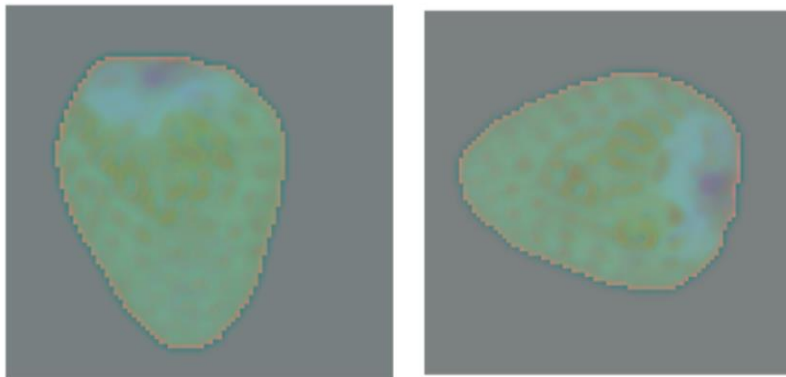


Fig 26. Imagen segmentada transformada en espacio de color LAB. (Fuente: Autor)

La clusterización mediante el algoritmo K-means en el espacio de color LAB permitió identificar y separar los diferentes tonos presentes en la imagen de la fresa, aquellos correspondientes a las regiones maduras e inmaduras. Se lograron agrupar los píxeles en clústers representativos, donde los tonos rojos y verdes se distinguen claramente como se muestra en la Fig 27.

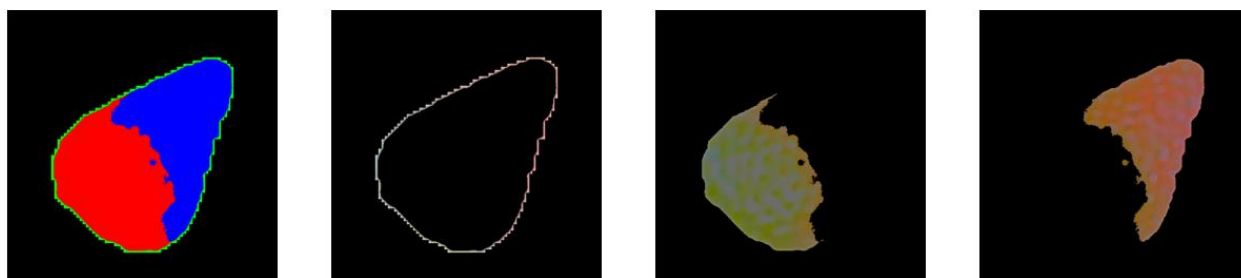


Fig 27. Clusterización de la imagen LAB. (Fuente: Autor)

Se procedió a cuantificar el porcentaje de maduración realizando un conteo de los píxeles en cada clúster, identificando aquellos correspondientes a los colores rojos y verdes. Las simulaciones demostraron que el análisis preciso de los clústers de color es fundamental para estimar correctamente la madurez, basándose en proporción de colores característicos de las fresas maduras e inmaduras.

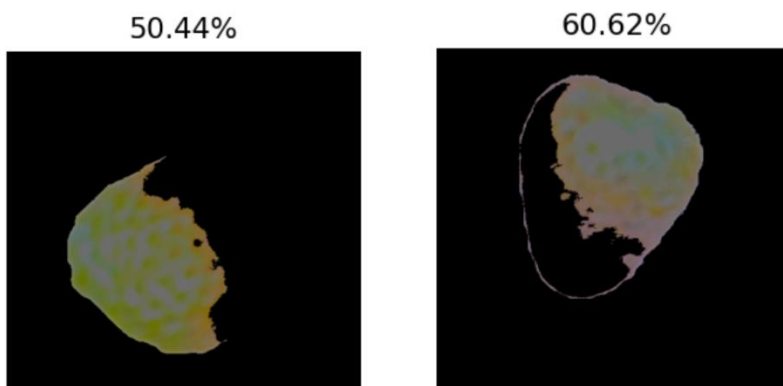


Fig 28. Porcentajes de inmadurez del fruto. (Fuente: Autor)

## 10. Conclusiones y Trabajos Futuros

A lo largo del trabajo, se evidenció que el desarrollo de un sistema automatizado para la selección de fresas según su madurez puede apoyarse de al menos tres áreas de la inteligencia artificial: visión por computador, Machine Learning y Deep Learning. Inicialmente, el proyecto se planteó con el objetivo de establecer una técnica para la clasificación de fresas. Durante el proceso de diseño, se abordaron diversos aspectos técnicos, siendo el principal el uso del algoritmo YOLO. Este algoritmo se empleó para la selección de fresas, basándose en clases preentrenadas que permiten una identificación y clasificación del fruto.

En este proyecto se llevó a cabo un estudio comparativo basado en el ajuste fino de YOLOv8 con sus variantes para la clasificación de fresas postcosecha, obteniendo resultados alentadores. Se lograron velocidades de procesamiento de 0.3 segundos utilizando el modelo entrenado. Estos resultados confirman que el modelo YOLOv8 puede clasificar las fresas en diferentes etapas de maduración en entornos complejos.

En cuanto a los parámetros visuales para determinar el grado de madurez del fruto, estos dependían en gran medida de la calidad de la imagen. Factores como la luminosidad influían directamente en el funcionamiento de los algoritmos, ya que afectaban la fidelidad de los datos adquiridos; desde la inteligencia artificial, se implementaron estrategias que permitieron adaptarse a las variaciones en la luminosidad. Por ejemplo, la técnica Retinex mitiga el impacto de la variable de luminancia (L) en la imagen capturada, ofreciendo además la ventaja de realzar el contraste. Esto facilita la identificación de características esenciales, como el color, dentro de la imagen.

A lo largo de la investigación se identificó que se debe contar con un sistema de adquisición de imagen de alta calidad, para resaltar las características requeridas del fruto. Esto es crucial porque la maduración del fruto no sigue un patrón predeterminado, por lo que los algoritmos deben extraer la mayor cantidad posible de características del fruto. Implementar un sistema de adquisición robusto permite que los algoritmos de inteligencia artificial operen con mayor precisión, adaptándose mejor a las variaciones intrínsecas del proceso de maduración del fruto.

La continuación de este proyecto se enfoca en la detección en tiempo real utilizando secuencias de video en lugar de imágenes estáticas. Este avance permitiría evolucionar el algoritmo de clasificación actual, integrando técnicas que identifiquen y rastreen frutos a lo largo de múltiples fotogramas. Abordar la detección en video ofrecería la capacidad de enfrentar desafíos como la variabilidad en la posición y orientación del fruto, así como los cambios en la iluminación, proporcionando una solución más adaptable para diferentes escenarios en la producción agrícola.

## 11. Referencias Bibliográficas.

- [1] D. N. Baker, J. R. Lambert, J. M. McKinion, and S. C. A. E. Station, *GOSSYM: A Simulator of Cotton Crop Growth and Yield*. in Technical bulletin (South Carolina Agricultural Experiment Station). S.C. Agricultural Experiment Station, 1983. [Online]. Available: <https://books.google.com.co/books?id=Y7SErgEACAAJ>
- [2] G. Bannerjee, U. Sarkar, S. Das, and I. Ghosh, “Artificial intelligence in agriculture: A literature survey,” *Int J Sci Res Comput Sci Appl Manag Stud*, vol. 7, no. 3, pp. 1–6, 2018.
- [3] N. C. Eli-Chukwu, “Applications of artificial intelligence in agriculture: A review,” *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, 2019.
- [4] T. M. Mitchell, *Machine Learning*. in McGraw-Hill International Editions. McGraw-Hill, 1997. [Online]. Available: <https://books.google.com.co/books?id=EoYBngEACAAJ>
- [5] Minagricultura, “Cadena de la Fresa,” *Dirección de Cadenas Agrícolas y Forestales*, Mar. 2021.
- [6] Gobernación de Boyacá, “Caracterización de la Producción Agrícola del Departamento de Boyacá,” *Boletín Red de Observatorios de Boyacá*, vol. 1, no. 4, 2022.
- [7] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach: International Edition*. Pearson Education, 2015. [Online]. Available: <https://books.google.com.co/books?id=pAWpBwAAQBAJ>
- [8] V. Por Computador Editado, E. Alegre, G. Pajares, and A. De La Escalera, “Conceptos y Métodos en,” 2016.
- [9] L. Enrique Sucar and M. Giovani Gómez, “Visión Computacional.”
- [10] M. Jordan, J. Kleinberg, and B. Schölkopf, “Pattern Recognition and Machine Learning.”
- [11] N. L. Montes Castrillón, “Segmentación de imágenes de frutos de café en el proceso de beneficio.” [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/2846>
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. in Adaptive Computation and Machine Learning series. MIT Press, 2016. [Online]. Available: <https://books.google.com.co/books?id=Np9SDQAAQBAJ>
- [13] J. Bobadilla, *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. Ediciones de la U, 2021. [Online]. Available: <https://books.google.com.co/books?id=iAAyEAAAQBAJ>
- [14] F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.
- [15] M. Z. Rodriguez *et al.*, “Clustering algorithms: A comparative approach,” *PLoS One*, vol. 14, no. 1, Jan. 2019, doi: 10.1371/journal.pone.0210236.
- [16] G. Zaccane and M. R. Karim, *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python, 2nd Edition*. Packt Publishing, 2018. [Online]. Available: <https://books.google.com.co/books?id=zZIUDwAAQBAJ>
- [17] Glenn Jocher and Muhammad Rizwan Munawar, “Ultralytics YOLO Docs,” Ultralytics Inc.
- [18] N. Ismail and O. A. Malik, “Real-time visual inspection system for grading fruits using computer vision and deep learning techniques,” *Information Processing in Agriculture*, vol. 9, no. 1, pp. 24–37, Mar. 2022, doi: 10.1016/j.inpa.2021.01.005.
- [19] H. Zhou, Z. Zhuang, Y. Liu, Y. Liu, and X. Zhang, “Defect classification of green plums based on deep learning,” *Sensors (Switzerland)*, vol. 20, no. 23, pp. 1–15, Dec. 2020, doi: 10.3390/s20236993.
- [20] D. Heras, “Clasificador de imágenes de frutas basado en inteligencia artificial,” *Killkana Técnica*, vol. 1, no. 2, p. 21, Nov. 2017, doi: 10.26871/killkana\_tecnica.v1i2.79.

- [21] P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna, and I. Escobar, "Artificial Vision Techniques to Optimize Strawberry's Industrial Classification," *IEEE Latin America Transactions*, vol. 14, no. 6, pp. 2576–2581, 2016, doi: 10.1109/TLA.2016.7555221.
- [22] J. Victor Aguilar-Alvarado and M. Alfredo Campoverde-Molina, "Classification of fruits based on convolutional neural networks Classificação de frutos com base em redes neurais convolucionais Ciencias de la ingeniería Artículo de investigación," vol. 5, no. 01, pp. 3–22, 2019, doi: 10.23857/pc.v5i01.1210.
- [23] E. De Postgrado, "UNIVERSIDAD PRIVADA ANTENOR ORREGO "ESPECTROSCOPIA CON INFRARROJO Y TECNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS-EMPRESA TalSA-2018 "."
- [24] A. Pajaziti, F. Basholli, and Y. Zhaveli, "Identification and classification of fruits through robotic system by using artificial intelligence," *Engineering Applications*, vol. 2, no. 2, pp. 154–163, May 2023, [Online]. Available: <https://publish.mersin.edu.tr/index.php/enap/article/view/974>
- [25] Y. Fonseca, C. Bautista, C. Pardo-Beainy, and C. Parra, "A plum selection system that uses a multi-class Convolutional Neural Network (CNN)," *J Agric Food Res*, vol. 14, p. 100793, 2023, doi: <https://doi.org/10.1016/j.jafr.2023.100793>.
- [26] J. P. Bonilla-González and F. A. Prieto-Ortiz, "Determinación del estado de maduración de frutos de feijoa mediante un sistema de visión por computador utilizando información de color," *REVISTA DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN*, vol. 7, no. 1, p. 111, Dec. 2016, doi: 10.19053/20278306.v7.n1.2016.5603.
- [27] S. Yang, W. Wang, S. Gao, and Z. Deng, "Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer," *Comput Electron Agric*, vol. 215, p. 108360, 2023, doi: <https://doi.org/10.1016/j.compag.2023.108360>.
- [28] A. Almutairi, J. Alharbi, S. Alharbi, H. F. Alhasson, S. S. Alharbi, and S. Habib, "Date Fruit Detection and Classification based on Its Variety Using Deep Learning Technology," *IEEE Access*, p. 1, 2024, doi: 10.1109/ACCESS.2024.3433485.
- [29] C. Dewi, O. M. Kamiasi, G. Chhabra, G. Dai, K. Kaushik, and I. U. Khan, "Automated Fruit Classification Based on Deep Learning Utilizing Yolov8," in *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2023, pp. 801–807. doi: 10.1109/UPCON59197.2023.10434542.
- [30] T. S. Gunawan, M. Kartiwi, H. Mansor, and N. M. Yusoff, "Palm Fruit Ripeness Detection and Classification Using Various YOLOv8 Models," in *2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2023, pp. 193–198. doi: 10.1109/ICSIMA59853.2023.10373435.
- [31] D.-L. Nguyen, X.-T. Vo, A. Priadana, M. D. Putro, and K.-H. Jo, "Fruit Ripeness Detector for Automatic Fruit Classification Systems," in *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*, 2024, pp. 1–6. doi: 10.1109/ISIE54533.2024.10595723.
- [32] C. Tang *et al.*, "A fine recognition method of strawberry ripeness combining Mask R-CNN and region segmentation," *Front Plant Sci*, vol. 14, 2023, doi: 10.3389/fpls.2023.1211830.
- [33] C. C. Tran, D. T. Nguyen, H. D. Le, Q. B. Truong, and Q. D. Truong, "Automatic dragon fruit counting using adaptive thresholds for image segmentation and shape analysis," in *2017 4th NAFOSTED Conference on Information and Computer Science*, 2017, pp. 132–137. doi: 10.1109/NAFOSTED.2017.8108052.

- [34] P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna, and I. Escobar, “Artificial Vision Techniques to Optimize Strawberry’s Industrial Classification,” *IEEE Latin America Transactions*, vol. 14, no. 6, pp. 2576–2581, 2016, doi: 10.1109/TLA.2016.7555221.
- [35] Y.-T. Kim, “Contrast enhancement using brightness preserving bi-histogram equalization,” *IEEE Transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1–8, 1997, doi: 10.1109/30.580378.
- [36] R. G. Apaza, C. E. Portugal-Zambrano, J. C. Gutiérrez-Cáceres, and C. A. Beltrán-Castañón, “An approach for improve the recognition of defects in coffee beans using retinex algorithms,” in *2014 XL Latin American Computing Conference (CLEI)*, 2014, pp. 1–9. doi: 10.1109/CLEI.2014.6965102.
- [37] L. Yang, D. Mu, Z. Xu, and K. Huang, “Apple Surface Defect Detection Based on Gray Level Co-Occurrence Matrix and Retinex Image Enhancement,” *Applied Sciences (Switzerland)*, vol. 13, no. 22, Nov. 2023, doi: 10.3390/app132212481.
- [38] D. Haba, *Data Augmentation with Python: Enhance deep learning accuracy with data augmentation methods for image, text, audio, and tabular data*. Packt Publishing, 2023. [Online]. Available: <http://ieeexplore.ieee.org/document/10251193>
- [39] J. T. Viñals, *Python Deep Learning : introducción práctica con Keras y TensorFlow 2*. Marcombo, 2020. [Online]. Available: <https://books.google.com.co/books?id=5vpmzQEACAAJ>
- [40] X. Guo, Y. Yang, C. Wang, and J. Ma, “Image dehazing via enhancement, restoration, and fusion: A survey,” *Information Fusion*, vol. 86–87, pp. 146–170, 2022, doi: <https://doi.org/10.1016/j.inffus.2022.07.005>.