Diseño de software para automatización del conteo de frutos mediante inteligencia y visión artificial.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Autores
Emily Castañeda Puentes
Sebastian Rivera Moreno

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
Facultad Tecnológica
Ingeniería en Control y Automatización

Bogotá D.C. 10 2024

Diseño de software para automatización del conteo de frutos mediante inteligencia y visión artificial



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Autores
Emily Castañeda Puentes 20211583020 emicastanedap@udistrital.edu.co
Sebastian Rivera Moreno 20211583003 sriveram@udistrital.edu.co

Modalidad
Monografía

Presentado para optar al título de: Ingeniero(a) en (Control y Automatización)

Director:
Frank Nixon Giraldo

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
Facultad Tecnológica
Ingeniería en Control y Automatización
Bogotá D.C. 10, 2024

Índice

Índice 3
Índice de Figuras 4
Índice de Tablas 4
Resumen 6
Abstract 6
1. Introducción 1
2. Planteamiento del problema 1
3. Justificación 4
4. Objetivos 5
    4.1. Objetivo General 5
    4.2. Objetivos específicos 5
5. Marco de referencia 6
    5.1. Antecedentes 6
    5.2. Marco teórico 8
    5.3. Marco Legal 10
6. Metodología 12
7. Resultados 13
    7.1. Propuesta definitiva 13
        Captura de imágenes en un entorno controlado: 13
        Base de datos 14
        Procesamiento de imágenes y entrenamiento del modelo 16
        Progreso del entrenamiento: 22
        Optimización y validación: 22
        Resumen del modelo: 22
        Resultados por clase: 23
        Ubicación de resultados: 24
        Validación de los resultados 24
        Desarrollo de la interfaz gráfica 25
    7.2. Restricciones de la solución 25
    7.3. validación de la solución (Resultados) 26
    7.4. Evaluación del impacto de la solución 32
8. Conclusiones 33
9. Referencias 34

Índice de Figuras

Figura 1. Diseño prototipo físico. ..……... 13
Figura 2. Prototipo en acrilico. ……. 14
Figura 3. Consultas sql. ..…….. 15
Figura 4. Entorno DBeaver. …….. 16
Figura 6. Delineado de frutos desde el software VoTT. …… 19
Figuras 7. Métricas del proyecto en VoTT. ……... 19
Figura 8. Configuración de exportación en VoTT. ……... 20
Figura 9. Formato YOLOv5. ……21
Figura 10. Formato YOLOv5. …….. 23
Figura 11. Interfaz ventana principal. ……... 25
Figura 12. Entrenamiento. …… 28
Figura 13. Validación Fuente: ……... 28
Figura 14. Results del entrenamiento y validación en métricas. ……...30
Figura 15. Matriz de confusión ……31
Figura 16. Resultados del conteo y detección de frutos. ……...32

Índice de Tablas

Tabla 1: Resultados por clase. …….. 23

Tabla 2: Interfaz gráfica. ……...27

Resumen

En la actualidad, la automatización y la inteligencia artificial están cambiando varias industrias, incluyendo la agricultura. Dado el incremento en la necesidad de eficiencia en la producción agrícola, es fundamental crear herramientas tecnológicas que mejoren labores clave como la medición instantánea de frutos en tiempo real. Esta necesidad surge en un contexto donde la precisión y rapidez son fundamentales para maximizar la producción y reducir costos.

El problema principal radica en la dificultad de realizar conteos manuales o semiautomáticos de frutos, lo cual es un proceso laborioso, propenso a errores humanos y costoso en términos de tiempo y recursos. Las soluciones tradicionales, basadas en el monitoreo visual manual o tecnologías de baja precisión, no son suficientes para atender las necesidades actuales del sector.

Para solucionar esta problemática, se ha diseñado un software basado en inteligencia artificial y visión artificial que permite automatizar el proceso de conteo de frutos en entornos agrícolas. Este sistema utiliza algoritmos avanzados de procesamiento de imágenes y técnicas de IA para identificar, clasificar y contar frutos con alta precisión, proporcionando resultados en tiempo real.

Los resultados obtenidos muestran que esta solución es eficiente en comparación con los métodos tradicionales. El tiempo de procesamiento se reduce significativamente, y la precisión en el conteo de frutos mejora, lo que en conjunto optimiza la producción agrícola y ayuda a reducir errores y desperdicios.

Este trabajo presenta una solución viable para la automatización del conteo de frutos, ofreciendo una alternativa más precisa y eficiente frente a las técnicas convencionales. La implementación de este sistema puede marcar una diferencia significativa en la productividad del sector agrícola y su competitividad en el mercado global.

Palabras clave: Automatización, Inteligencia artificial, Visión artificial, Agricultura de precisión.

Abstract

Currently, automation and artificial intelligence are transforming various industries, including agriculture. As the demand for efficiency in agricultural production processes increases, the development of technological tools that optimize essential

tasks, such as real-time fruit counting, becomes necessary. This need arises in a context where precision and speed are crucial to maximizing production and reducing costs.

The main problem lies in the difficulty of performing manual or semi-automatic fruit counting, which is a laborious process, prone to human error, and costly in terms of time and resources. Traditional solutions, based on manual visual monitoring or low-precision technologies, are not sufficient to meet the current needs of the sector.

In response to this issue, a software based on artificial intelligence and computer vision has been developed to automate the fruit counting process in agricultural environments. The system uses advanced image processing algorithms and IA techniques to identify, classify, and count fruits with high accuracy, providing real-time results.

The results obtained show that this solution is much more efficient than traditional methods. Processing time is significantly reduced, and the accuracy in fruit counting improves, which optimizes agricultural production overall and helps reduce errors and waste.

In conclusion, this work presents a viable solution for automating fruit counting, offering a more precise and efficient alternative compared to conventional techniques. The implementation of this system can make a significant difference in the productivity of the agricultural sector and its competitiveness in the global market.

Keywords: Automation, Artificial Intelligence, Computer Vision, Precision Agriculture.

1. Introducción

Actualmente la agricultura enfrenta el reto de aumentar la eficiencia y productividad para satisfacer la creciente demanda global de alimentos. En este contexto, la integración de tecnologías avanzadas, como la inteligencia artificial (IA) y la visión artificial, ha comenzado a transformar procesos tradicionalmente manuales en soluciones automatizadas que optimizan el uso de recursos. La automatización agrícola no solo mejora la precisión y rapidez de tareas críticas, sino que también reduce la dependencia de la mano de obra intensiva, lo que es particularmente importante en un entorno donde los costos y la disponibilidad de personal cualificado son variables significativas. [1]

Uno de los procesos más esenciales dentro de la agricultura moderna es el conteo preciso de frutos. Este paso es fundamental para el monitoreo de la productividad, el control de calidad y la toma de decisiones en las etapas de cosecha y distribución. Los métodos tradicionales de conteo, que suelen implicar inspecciones visuales manuales, son propensos a errores humanos, consumen mucho tiempo y recursos, y a menudo no logran cumplir con los niveles de precisión que la industria requiere en la actualidad.

En respuesta a estas limitaciones, este trabajo presenta el desarrollo de un software basado en inteligencia artificial y visión artificial que automatiza el proceso de conteo de frutos. El sistema es capaz de identificar y contar frutos con alta precisión y en tiempo real. El objetivo de este proyecto es demostrar cómo la implementación de este tipo de tecnologías puede contribuir a una mayor eficiencia y precisión en el sector agrícola, mejorando tanto la productividad como la competitividad en un mercado global cada vez más exigente.

2. Planteamiento del problema

La gestión eficiente de los cultivos de frutas es de vital importancia para agricultores, investigadores y productores agrícolas, ya que influye directamente en la productividad y en la planificación de la cosecha. Para lograr una producción eficiente y rentable, es esencial contar con información precisa sobre el estado y rendimiento de los cultivos. Entre las tareas más críticas está la cuantificación precisa de los frutos producidos, ya que esta información es fundamental para la planificación de la cosecha, la evaluación de la productividad y la toma de decisiones comerciales.

Actualmente, el conteo de frutos suele realizarse de manera manual o utilizando técnicas rudimentarias, lo que genera importantes limitaciones en términos de precisión y eficiencia. En cultivos extensos, el proceso de conteo manual no solo es tedioso y laborioso, sino que también es propenso a errores humanos, lo que puede llevar a resultados inexactos y afectar negativamente la gestión de la producción. Estos errores pueden ser causados por fatiga, condiciones ambientales adversas o simplemente por la complejidad de la tarea. Como resultado, la información obtenida a partir de este proceso puede no ser lo suficientemente confiable para tomar decisiones estratégicas, lo que puede impactar negativamente la planificación de la cosecha y, en última instancia, la rentabilidad del productor.

Además, la dependencia de la mano de obra calificada para realizar estas tareas es otro factor problemático. El conteo manual requiere personal entrenado que pueda llevar a cabo el proceso de manera eficiente, lo que no solo representa un costo considerable en términos de recursos humanos, sino que también está sujeto a la disponibilidad de mano de obra en temporadas de alta demanda. En muchas regiones agrícolas, la escasez de trabajadores capacitados es un desafío creciente, lo que agrava aún más las dificultades de realizar el monitoreo de manera precisa y oportuna. [2]

Por otro lado, la utilización de métodos tradicionales limita la posibilidad de recopilar datos en tiempo real, lo que impide una rápida adaptación a las condiciones cambiantes del cultivo o del mercado. En un entorno donde la agricultura de precisión y el uso de tecnologías avanzadas están transformando la forma en que se gestionan los cultivos, las técnicas manuales de conteo de frutos se han vuelto insuficientes para atender las demandas actuales de precisión, eficiencia y escalabilidad.

Ante esta problemática, surge la necesidad de desarrollar herramientas tecnológicas que permitan automatizar el proceso de conteo de frutos, proporcionando una

alternativa más precisa, rápida y rentable. Estas herramientas no solo deben ser capaces de operar en tiempo real, sino que también deben integrarse de manera eficiente en los sistemas actuales de gestión agrícola para maximizar la productividad y reducir costos.

3. Justificación

La necesidad de contar con herramientas tecnológicas avanzadas para mejorar la eficiencia y precisión en la gestión de los cultivos es cada vez más urgente, especialmente en un contexto donde la agricultura de precisión está en auge. Los métodos tradicionales de conteo manual, como se ha mencionado previamente, presentan limitaciones considerables tanto en términos de tiempo como de exactitud. Estas limitaciones afectan directamente la capacidad de los agricultores y productores para tomar decisiones informadas en torno a la productividad, la planificación de la cosecha y la optimización de los recursos. Para mitigar estos desafíos, es necesario el desarrollo de soluciones tecnológicas que automatizan el proceso de conteo de frutos de manera precisa y eficiente.

El desarrollo de un software basado en visión artificial y procesamiento de imágenes responde a esta necesidad, proporcionando una alternativa automatizada que reduce la dependencia de métodos manuales propensos a errores humanos. A través de tecnologías de inteligencia artificial, es posible analizar imágenes capturadas en el campo o al momento de cargue para transporte, identificar automáticamente los frutos, contarlos con precisión y ofrecer información detallada sobre la cantidad de frutos. Este tipo de soluciones no solo ahorran tiempo, sino que también aumentan significativamente la precisión del conteo, eliminando errores derivados de la fatiga o el error humano.

Adicionalmente, la implementación de este software no solo permitirá mejorar la precisión del conteo de frutos, sino que también ofrecerá una visualización en tiempo real desde la web. La variabilidad en el color, forma y disposición de los frutos, así como las diferentes condiciones de iluminación y fondo, son aspectos clave que deben ser considerados en el diseño del software para garantizar su aplicabilidad en el entorno de empaque. La capacidad del software para manejar estas variaciones garantizará su utilidad y fiabilidad.

También, la automatización del conteo de frutos mediante el uso de visión artificial no solo responde a una necesidad actual del sector agrícola, sino que también ofrece una solución escalable que puede mejorar la eficiencia operativa y reducir costos a largo plazo. La integración de este tipo de tecnología en los procesos agrícolas puede marcar una diferencia significativa en la competitividad de los productores agrícolas en un mercado global en constante evolución.

4. Objetivos

4.1. Objetivo General

1. Diseñar un software para automatizar el conteo de frutos con inteligencia y visión artificial.

4.2. Objetivos específicos

2. Desarrollar un prototipo para captura de imágenes en un entorno controlado.

3. Implementar una base de datos para el almacenamiento de la información.

4. Diseñar un algoritmo para el procesamiento de las imágenes y entrenamiento del modelo para el reconocimiento de frutos

5. Validar los resultados del modelo de reconocimiento de frutos

6. Desarrollar una interfaz gráfica para manejo del software

5. Marco de referencia

5.1. Antecedentes

En el artículo “Software para la automatización de conteo de plantas en cultivos de papa mediante visión artificial” tuvo como objetivo implementar y evaluar diferentes métodos para el conteo de individuos en cultivos de papa, mediante visión artificial, enfocados a realizar un inventario digital y validarlo con los datos tomados en diferentes etapas del cultivo. Realizaron la adquisición de imágenes en una finca en Zipaquirá por medio de un dron con sensores adecuados para que la captura de imágenes fuera útil para realizar el procesamiento, adicional hicieron tomas a distintas alturas y en distintas etapas de maduración del cultivo donde se determinó que a menor altura tiene mayor veracidad, una vez determinaron la adquisición como correcta eligieron realizar una segmentación en las imágenes procesadas con tres métodos; umbralización, espacio de color y K-medidas, lo que permitió aplicar un bloque de refinamiento para mejorar los resultados y obtener una máscara para mantener los bordes de la imagen y poder hacer el conteo de las planta por medio de los contornos externos de las plantas o el material vegetal detectado en cada uno de los surcos. [3]

En el artículo “Sistema de Conteo Forestal, Agrícola y Ganadero mediante procesamiento de imágenes” realizaron un software para el conteo de plantas y ganado con procesamiento digital de videos tomados desde un dron aplicando visión artificial, estos videos fueron tomados en distintas alturas y distintas condiciones ambientales, finalmente el prototipo creado fue exitoso proponiendo un dispositivo comercial con mayor variación en el momento de detección de objetos y con bases de datos amplias adecuadas para varios tipos de campo y condiciones ambientales. [4]

En el artículo “Machine vision system for weed detection using image filtering in vegetable crops” expusieron un sistema de visión que se implementara en un robot móvil removedor de maleza, este tendrá la cámara adecuada para control de luz y obtener las imágenes exteriores. En el proyecto se desarrolló un algoritmo entrenado para el reconocimiento de maleza usando filtros en la imagen y extrayendo características de color y área, luego se implementaron un proceso de etiquetado a casa objeto de la escena. Concluyeron que una clasificación y tener un área delimitada para trabajo es más fácil de procesar, teniendo en cuenta los datos de especificidad, valores predictivos positivos y negativos de las imágenes. [5]

En el artículo “Comparative Analysis between LDR and HDR Images for Automatic Fruit Recognition and Counting” realizaron un análisis comparativo

entre imágenes LDR y HDR, para la detección y conteo de frutos, concluyendo que las imágenes HDR tienen una mejora del 30% en comparación con LDR en la detección. Ejecutaron distintas técnicas de tonal, en una imagen obtuvieron tres exposiciones de una misma escena, una con exposición normal equivalente a la imagen LDR, una sobreexpuesta y otra subexpuesta, estas se combinaron produciendo una imagen HDR. Para este punto aplicaron tres algoritmos de mapeo de tonos desde Matlab, les dio como resultado cuatro imágenes, estas se analizan con un algoritmo de detección y conteo de frutas donde pueden obtener el número de frutos correctamente identificados, los falsos positivos y el número total de frutos. Adicional como utilizaron funciones de Matlab fue necesario definir seis condiciones para que fuera efectiva la ejecución del código; los límites mínimo y máximo de los componentes de tono, el área bajo la cual se eliminarán las regiones, el radio máximo y mínimo de los círculos a detectar y la sensibilidad de la función de detección de círculos. [6]

En el artículo "Sistema automático de reconocimiento de frutas basado en visión por computador" se presenta un sistema capaz de reconocer fruta tropical latinoamericana por medio de una base de datos y usando técnicas de visión por procesamiento de imágenes. Realizaron segmentación mediante umbral sobre una capa de color azul del modelo RGB esto siento posible ya que en el prototipo para la toma de imágenes acoplaron un fondo blanco. En la capa azul se genera alto contraste contra el fondo haciendo que la obtención de segmentación de la fruta sea óptima utilizando un valor de umbral bimodal, luego de esto utilizaron binarización global de la capa azul con una función de Matlab (im2bw), esta función transforma la imagen azul en un modelo RGB dando como resultado una nueva imagen binaria (1 y 0) en base al umbral calculado. De esta nueva imagen con unos y ceros, se representa muy bien la posición de la fruta permitiendo la obtención de la mascara que permite identificar y eliminar los pixeles de fondo, ya para el reconocimiento de frutas utilizaron un clasificador KNN y de Bayes utilizando combinaciones y características HSV y RGB obteniendo porcentaje de acierto diferentes pero superior en el caso de KNN y RGB con el 90% de acierto. [7]

En el artículo "conteo de flores y frutos para el monitoreo del cultivo de aguacate Hass por medio de imágenes utilizando Machine Learning" muestran el desarrollo de una aplicación que por medio del procesamiento de imágenes de baja resolución se logra hacer conteo automático de las flores y los frutos en los árboles, hicieron uso de una red neuronal YOLOv5 con la cual obtuvieron 80% de precisión con los aguacates y 39% para las flores, esta red cuenta con 24 capas convolucionales seguidas de dos capas completamente conectas, al ser así, es capaz de detectar el contexto de la imagen y los objetos en ella, este proyecto demostró que el

monitoreo de los cultivos puede ser de bajo costo obteniendo buenos resultados de reconocimiento. [8]

En la revista Automation in Agriculture by Machine and Deep Learning Techniques se presentó un artículo llamado “A Review of Recent Developments” mostrando que el uso de sistemas de visión artificial en la agricultura puede mejorar significativamente la precisión en tareas como la clasificación y el conteo de frutos, proporcionando resultados en tiempo real. Esto es crucial para mejorar la toma de decisiones en la gestión de los cultivos. Además, la integración de algoritmos de aprendizaje profundo permite que los sistemas de conteo automaticen la identificación de variaciones en el tamaño, forma y color de los frutos, incluso bajo condiciones ambientales complejas. [9]

El estudio "Vegebot" publicado en The Journal of Field Robotics desarrollado por ingenieros de la Universidad de Cambridge, este robot utiliza visión artificial y algoritmos de aprendizaje automático para identificar y cosechar lechugas tipo iceberg. Aunque todavía no alcanza la velocidad de los trabajadores humanos, su potencial radica en reducir la mano de obra manual, abordar la escasez de trabajadores y minimizar el desperdicio de alimentos. El sistema cuenta con cámaras que identifican qué cultivos están listos y optimiza su cosecha mediante ajustes precisos en la presión de las herramientas de corte para evitar daños a los vegetales. [10]

5.2. Marco teórico

Visión artificial: La visión artificial, también conocida como visión por computadora o visión de máquina, es un campo interdisciplinario que combina conocimientos de ingeniería, matemáticas, ciencias de la computación y neurociencia para desarrollar sistemas informáticos capaces de interpretar y comprender el contenido visual de imágenes o vídeos. Es decir, la visión artificial se refiere a la capacidad de las máquinas para "ver" y comprender el mundo visual que la rodea de manera similar a los seres humanos. Esto implica el desarrollo de algoritmos y técnicas de procesamiento de imágenes que permiten a las computadoras analizar y extraer información útil de datos visuales. Algunas de las tareas comunes en visión artificial incluyen:

- Detección de objetos: Identificar la presencia y la ubicación de objetos específicos en una imagen o vídeo, como vehículos, personas, edificios, etc.
- Reconocimiento de patrones: Identificar y clasificar patrones específicos en datos visuales, como reconocimiento facial, reconocimiento de caracteres, etc.

- Seguimiento de movimiento: Seguir y rastrear el movimiento de objetos en una secuencia de imágenes o vídeo a lo largo del tiempo.
- Segmentación de imágenes: Dividir una imagen en regiones o segmentos basados en características visuales, como color, textura, forma, etc.
- Estimación de profundidad: Determinar la distancia entre la cámara y los objetos en una escena, proporcionando información tridimensional sobre la escena visual. En la agricultura, la visión artificial tiene una amplia gama de aplicaciones, como monitoreo del crecimiento de cultivos, detección de enfermedades en plantas, clasificación de frutas y verduras según su tamaño, forma o madurez, y conteo de plantas en campos agrícolas. [11]

Inteligencia artificial: La inteligencia artificial o IA se puede describir como la capacidad de las máquinas para simular inteligencia humana, es decir, capacidad para aprender, razonar, percibir, planificar y tomar decisiones. Está basada en la creación de algoritmos y sistemas informáticos completos que pueden llegar a procesar grandes cantidades de datos y realizar la tarea asignada de manera autónoma. En el campo de la informática que se enfoca en el desarrollo de sistemas informáticos capaces de realizar tareas que normalmente requieren inteligencia humana, como el reconocimiento de patrones, el procesamiento del lenguaje natural, la toma de decisiones. [12]

Base de datos: Una base de datos es un sistema organizado de almacenamiento de datos que permite el acceso, la recuperación, la manipulación y la gestión eficiente de la información. Se compone de un conjunto de datos interrelacionados, organizados y estructurados de manera que sea fácil de consultar y actualizar. [13]

Red Neuronal: Una red neuronal es un modelo computacional inspirado en el funcionamiento del sistema nervioso biológico, es decir, el cerebro humano, diseñado para procesar y aprender información a partir de datos. Consiste en una red interconectada de nodos o neuronas artificiales que trabajan en conjunto para realizar una tarea en específico, se puede aplicar en el reconocimiento de patrones, clasificación de datos, predicciones entre otros. [14]

OpenCV: Open Source Computer Vision Library es una biblioteca de código abierto diseñada para la visión por computadora y el procesamiento de imágenes. Proporciona una amplia gama de funciones y algoritmos para el análisis de imágenes y vídeos, incluyendo procesamiento de imágenes, detección de objetos, reconocimiento facial, seguimiento de movimiento, calibración de cámaras, entre otros. [15]

TensorFlow: Es una biblioteca de software de código abierto desarrollada por Google Brain Team que se utiliza para implementar y desplegar modelos de aprendizaje automático y redes neuronales. TensorFlow proporciona una interfaz flexible y escalable para la construcción y entrenamiento de modelos de aprendizaje automático, así como herramientas para su despliegue en diferentes entornos. [16]

5.3. Marco Legal

Ley de Software

- Ley 603 de 2000: Esta ley tiene como objetivo principal promover el desarrollo de la industria del software en Colombia. Establece disposiciones relacionadas con la propiedad intelectual, la protección de los derechos de autor y el fomento de la investigación y la innovación en el campo del software.

Regulaciones de ciberseguridad y software

- Decreto 1074 de 2015: Este decreto reglamenta el sector de tecnologías de la información y las comunicaciones en Colombia. Aunque no se centra exclusivamente en ciberseguridad, contiene disposiciones relevantes para la protección de la información y la seguridad de los sistemas informáticos.
- Estándares de Calidad de Software: Aunque no hay regulaciones específicas en Colombia, las empresas de software pueden seguir estándares internacionales reconocidos, como ISO/IEC 25010 sobre calidad de productos de software.
- Reglamentación de Inteligencia Artificial: Aunque aún no existe una regulación específica sobre inteligencia artificial en Colombia, es importante estar al tanto de las discusiones y desarrollos en este campo a nivel nacional e internacional, ya que pueden influir en el desarrollo y la regulación futura de tecnologías como la visión e inteligencia artificial.

Normativas agrícolas

Buenas Prácticas Agrícolas (BPA):

- Resolución 4579 de 2012: Esta resolución establece las normas técnicas y los requisitos para la implementación de las Buenas Prácticas Agrícolas en Colombia. Las BPA son un conjunto de principios y prácticas que promueven la producción agrícola sostenible, la protección del medio ambiente y la seguridad alimentaria.

Calidad de los alimentos

- Ley 118 de 1994: Esta ley establece disposiciones para garantizar la calidad, inocuidad y etiquetado adecuado de los alimentos en Colombia. La ley establece estándares y requisitos para la producción, procesamiento, almacenamiento, transporte y comercialización de alimentos, con el fin de proteger la salud pública y promover la transparencia en la información al consumidor.

Reglamento del sector agropecuario

- Decreto 1071 de 2015: Reglamentario del Sector Agropecuario: Este decreto reglamenta varios aspectos relacionados con el sector agropecuario en Colombia, incluyendo la producción agrícola, ganadera, pesquera y forestal. El decreto establece disposiciones sobre el registro de predios agrícolas, el uso sostenible de los recursos naturales, la conservación del suelo y el agua, entre otros.

6. Metodología

En la fase uno se identificó la necesidad de contar frutos de manera automatizada en el entorno agrícola, estableciendo los objetivos del proyecto, como la precisión en el conteo, la velocidad de procesamiento y la facilidad de uso.
Se realizó un bosquejo del prototipo, determinando la parte física y mecánica del sistema, así como los requerimientos electrónicos.

En la fase dos, se llevó a cabo la investigación de las características deseadas del software, como la compatibilidad con la cámara web, la capacidad de procesar imágenes en tiempo real y la integración de la Raspberry pi.

En la tercera fase se realizó el análisis y selección de tecnologías, lenguaje de programación como Python, OpenCV para el procesamiento de imágenes y TensorFlow para la implementación de redes neuronales.

De cuarta fase, se hizo el diseño de la arquitectura del sistema, incluyendo la estructura del software, la integración de la cámara web y la Raspberry Pi, y la implementación de algoritmos de visión e inteligencia artificial.

En la quinta fase se realizó la de implementación del software haciendo la programación de los algoritmos de procesamiento de imágenes y redes neuronales para el conteo de frutos y la adecuación de la Raspberry con la cámara al software.

En la sexta fase, se realizaron las pruebas y análisis de resultados, haciendo el entrenamiento del sistema con la base de datos y se verificó la funcionalidad y rendimiento, se realizó pruebas en distintas condiciones de iluminación y ángulo de la cámara con los distintos frutos.

En la séptima y última fase, se realizó el documento escrito que da a conocer el desarrollo del prototipo.

7. Resultados

7.1. Propuesta definitiva

Captura de imágenes en un entorno controlado:

Se propone como punto de partida realizar un esquema del prototipo físico para la captura de imágenes donde va el sistema de visión y la Raspberry Pi 3 B, como se muestra en la figura 1.

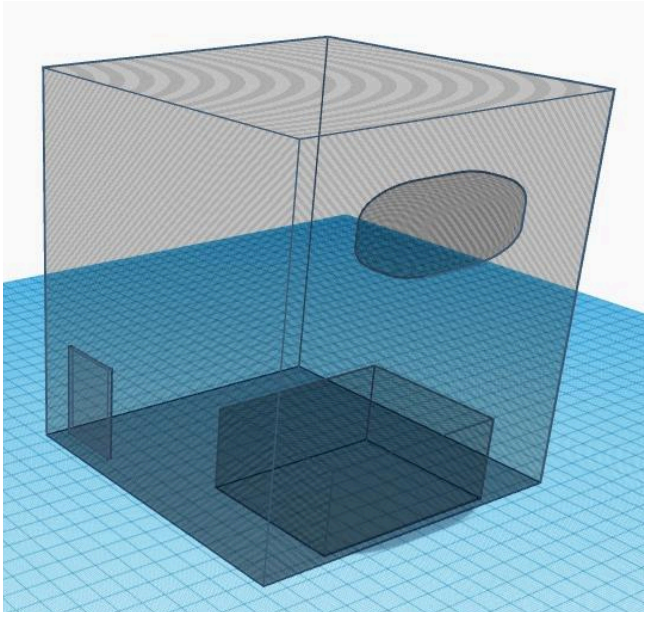


Figura 1. Diseño prototipo físico. *Fuente: Autores*

Una vez realizado este diseño se elige como material acrilico negro quedando como se muestra en la figura 2.

Se diseñó y construyó un prototipo capaz de capturar imágenes en un entorno controlado. Este prototipo incluye cámara Logitech C505 HD que puede ajustar parámetros de captura de imágenes según las condiciones del entorno, como la luz y la distancia. Se trabajó con configuraciones que permiten obtener imágenes precisas, incluso con diferentes condiciones de iluminación y tipos de frutos (bananos, limones, lulos, mandarinas y naranjas).

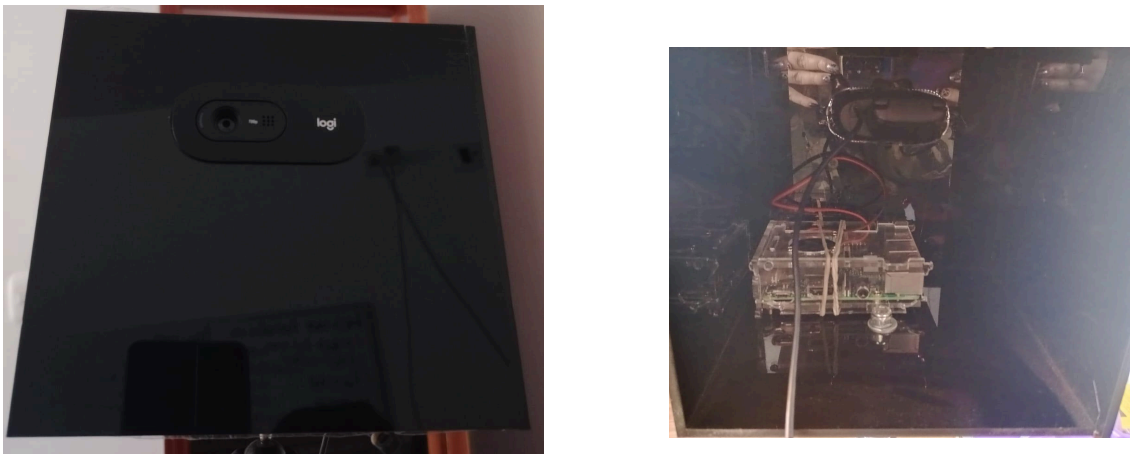


Figura 2. Prototipo en acrilico. *Fuente: Autores*

NOTA: Para la realización de la propuesta definitiva se tuvo en cuenta que como requerimiento principal se iba a tener que hacer uso de GitHub como forja, en el equipo en el que se iba a trabajar, así mismo se debe instalar en la Raspberry pi**.**

Base de datos

Para manejar el flujo de datos generado por el sistema, se implementó una base de datos robusta desde MySQL con lenguaje sql administrada desde el software DBeaver, con el cual se gestionan los usuarios, se realiza la creación de tablas y cualquier otro tipo de consulta sql (ver figura 3) para el manejo de la información, en este software se puede visualizar de manera externa a la interfaz gráfica el registro de eventos, (conteo de frutos, inicio y cierre de sesión de usuario, falla de conteo de frutos), también se puede almacenar el registro de los procesamientos hechos, guardar la cantidad de frutos que contó, recibe la fecha y qué usuario ingresó, esto contenido en las diferentes tablas creadas.

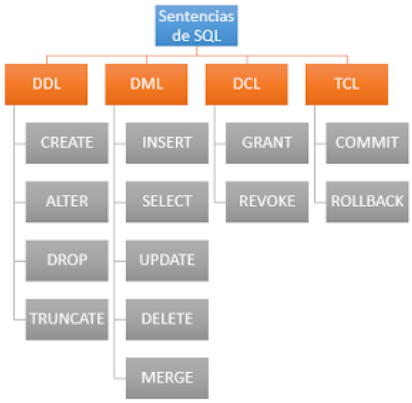


Figura 3. Consultas sql. *Fuente:*

En la figura anterior se visualizan algunas de las consultas que se pueden realizar en sql. En la base de datos trabajada se implementó consulta DDL, especial para crear tablas y organizar relaciones entre ellas, DML, utilizada para la manipulación de datos y DCL, para la gestión de datos, múltiples tablas o vistas.

Así bien, con DDL se implementó *CREATE,* para la creación de la base de datos, con DML se utilizaron *INSERT* para insertar un registro en una tabla, *SELECT* para seleccionar diferentes registros dependiendo el criterio, *UPDATE* para actualizar el registro, *DELET* para borrar el registro y finalmente con DCL se utilizó *GRANT* para darle los permisos al usuario que se creó.

Teniendo en cuenta lo anterior se realiza desde MySQL se usa la terminal command line client, para la creación de la base de datos y creación de tablas y campos.

Una vez realizado la creación de la base de datos con todos los elementos se hace uso de ella mediante mediante consultas sql definidas en el código de la aplicación.

En la figura 4 se puede visualizar el software DBeaver, con una de las tablas, correspondiente a la de usuarios, en la parte izquierda de la imagen se puede apreciar el árbol de trabajo donde está el acceso a las otras tablas, eventos, imágenes, registros y users.

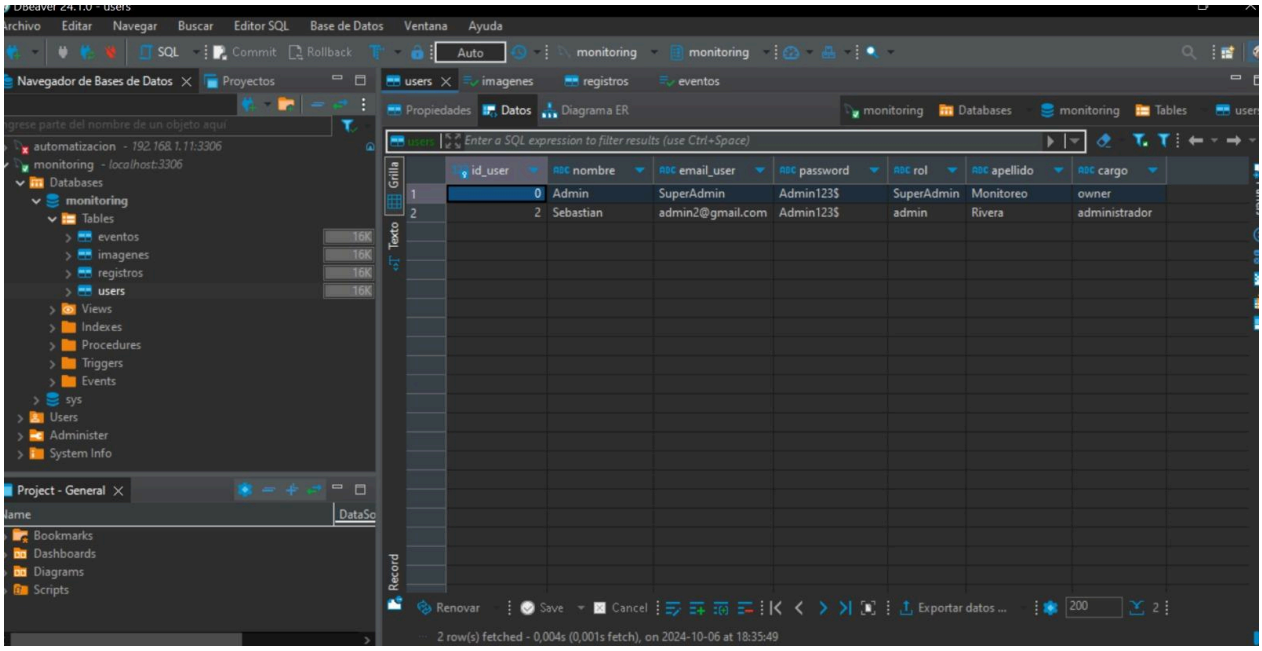


Figura 4. Entorno DBeaver. *Fuente: Autores*

La base de datos permite tener el registro histórico de los eventos ingresados en la plataforma.

Procesamiento de imágenes y entrenamiento del modelo

Se implementó un algoritmo de procesamiento de imágenes para reconocimiento de objetos que está basado en IA. El algoritmo identifica objetos en imágenes, para el proyecto se generó un dataset con imágenes de los frutos que queremos identificar. Tomando cada imágen se realizó un etiquetado de los frutos que se utilizaron para el entrenamiento del modelo.

Para el entrenamiento del modelo se utilizó YOLOv5s, un modelo de detección de objetos por IA, con el dataset de las imagénes tomadas y las librerías de python tensorflow, pillow, torchvision, se creo un modelo personalizado para la detección de frutos seleccionados.

Como software principal de editor de código se utilizó Visual Studio Code ya que permite el uso de todo el repositorio de manera organizada y eficaz, además permite una amplia variedad de extensiones que nos ayudan a visualizar los errores de los diferentes lenguajes de programación utilizados (Python, JS, HTML y CSS).

Teniendo en cuenta la información anterior para el preprocesamiento y entrenamiento del modelo se realizaron los siguientes pasos.

1. Desarrollo de un algoritmo para captura de imágenes y guardarlas etiquetadas en base a la fruta que se tomó.

Se utilizó OpenCV que es una herramienta para realizar operaciones de procesamiento de imágenes, como el filtrado, la detección de bordes, el reconocimiento de características, el seguimiento de objetos, etc. Así bien se creó un script en el cual se abría una ventana donde se visualiza lo que se estaba obteniendo de la cámara, se procede con la captura de las imágenes y estas se iban guardando en orden con el nombre y el número con el que iba la cuenta de imágenes en una carpeta, estas imágenes se guardaron en formato .jpg, este proceso se realizó 300 veces por tipo de fruto en distintas condiciones de iluminación, distancia de la cámara, posición y cantidad del mismo fruto en la misma imágen, finalizando el proceso se cerraba el script para luego volver a abrirlo pero cambiando la etiqueta del siguiente fruto al capturar; los frutos utilizados en el proyecto bananos, limones, lulos, naranjas y mandarinas para un total de 1500 imágenes.

2. Entrenar el modelo

Para entrenar el modelo utilizando YOLOv5 que funciona mediante etiquetas, se definen las regiones en donde están los objetos que se quiere identificar y luego a esas regiones se les asigna una clase, en este caso las regiones corresponden a los frutos que seleccionamos y se guardaron de la siguiente manera:
0 lemon
1 tangerine
2 orange
3 banana
4 lulo

Siendo el número correspondiente a la clase y la región el nombre del fruto.

Se utilizó un software de código abierto de entrenamiento de etiquetado de objetos en imágenes llamado VoTT (Visual Object Tagging Tool), y se iban guardando las imágenes en una carpeta, luego se cargó el repositorio dentro de este software y se empezó con el etiquetado y delineado de cada fruto como se visualiza en la figura

6, este proceso se realizó con las 1500 imágenes tomadas inicialmente y fruto por fruto.

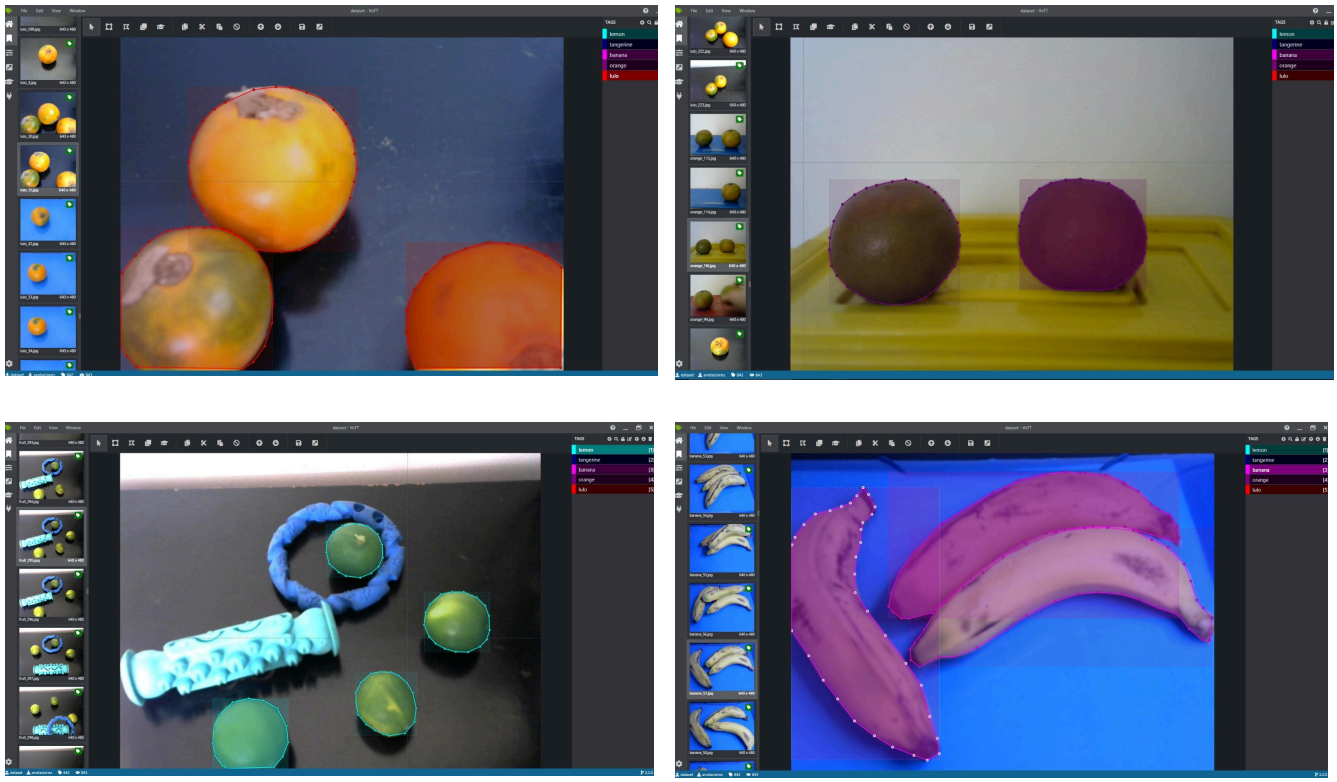


Figura 6. Delineado de frutos desde el software VoTT. *Fuente: Autores*

Este software crea un nuevo archivo llamado *anotaciones*, en este archivo están las coordenadas de los objetos delineados con las etiquetas que se van a usar.

Nota: Se obtiene un dato diferente de regiones a número de imágenes por etiqueta, ya que en algunas fotos había más de un fruto, así que en la figura 7 donde se visualizan las métricas por fruto podemos ver que el número de regiones delineadas es mayor al número de imágenes.



Figuras 7. Métricas del proyecto en VoTT. *Fuente: Autores*

Este software permite realizar una exportación de distintos tipos de conjunto de datos como se muestra en los figura 8, para el proyecto se exportó de tipo pascal VOC, que es un conocido conjunto de datos de detección, segmentación y clasificación de objetos, adicional que al exportar genera unas anotaciones en la ruta que se elija, en este caso en el repositorio del proyecto en la dirección de visual studio code.

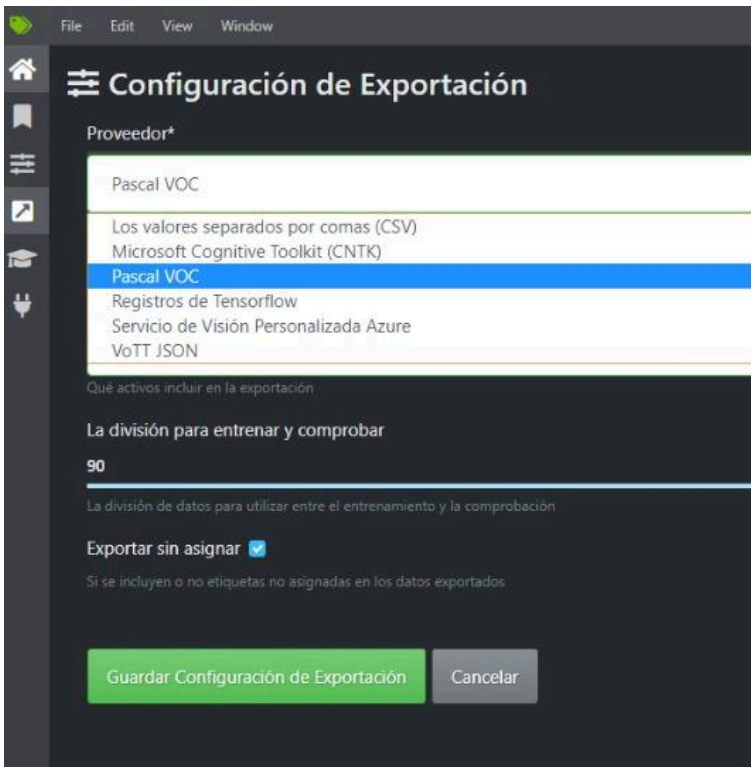


Figura 8. Configuración de exportación en VoTT. *Fuente: Autores*

Una vez creada la exportación en la ruta deseada, se importa desde visual studio code, se generan unas anotaciones en formato xml. que tienen la información de origen (donde están las imágenes) y donde se van a guardar las anotaciones que se van generando.

En este caso el formato con YOLOv5 entraba en conflicto, como se visualiza en la figura 9, la manera en la que se generan los datos de una imágen es en orden de; clase, coordenada centro en x, coordenada centro en y, ancho y alto. Cómo generamos la importación con formato pascal VOC que nos entrega coordenada min en x, coordenada max en x, coordenada min en y, y coordenada max en y , como solución se crea un script para convertir de pascal VOC a YOLOv5.

Este script se realiza en python, lee los archivos .xml, se especifica dónde encontrar las anotaciones, donde se va a guardar en formato YOLOv5 y qué clases de frutos se tienen ya que en formato pascal VOC nos aparece por ejemplo “lemon” pero en formato de anotaciones tiene que ser un índice de la RAE, luego se convierten las dimensiones, se ejecuta y se genera un nuevo archivo. Este archivo nos dice el índice, los datos de coordenada centro en x, coordenada centro en y, ancho y alto.

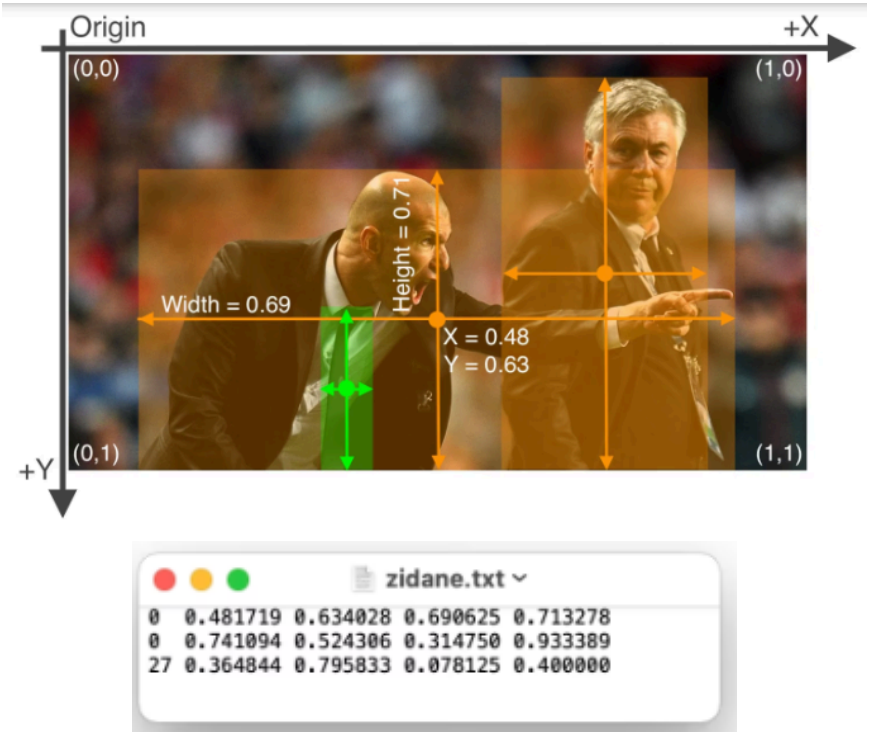


Figura 9. Formato YOLOv5. *Fuente: Ultralitycs; entrenar datos personalizados*

Luego de ejecutar este script y convertidos los datos a YOLOv5 exitosamente se organizó el directorio para entrenar el modelo:

- Directorio central con una carpeta que se llame imagen (nombre de la imagen debe ser igual al nombre de la anotación)
- Se crean dos archivos .txt (son archivos de entrenamiento y validación). El archivo de entrenamiento es el correspondiente a adquisición de anotaciones de 1200 imágenes, y el de validación toma un número determinado de imágenes (en este caso 300) y realiza validación.
  Para esta elección de imágenes también se crea un script que seleccione al azar 1200 imágenes definiendo el 80% de cada archivo de las imágenes (es decir de cada fruto toma 250 imágenes y las almacena para archivo de entrenamiento) y las 50 imágenes restantes de cada archivo las asigna para el archivo de validación.

Para completar el entrenamiento una vez generados los dos archivos .txt, trabajamos la carpeta de YOLOv5 que tiene toda la copia del repositorio y se ejecuta un script creado que solicita unos parametros especificos:

--img 640: El tamaño de las imágenes es 640x640 píxeles.
--batch 16: El tamaño del lote (batch size) es de 16 imágenes por iteración.

--epochs 100: El entrenamiento se realizó durante 100 épocas (o ciclos completos de entrenamiento).
--data frutas.yaml: Este archivo YAML contiene la configuración del conjunto de datos (por ejemplo, clases e imágenes).
--cfg models/yolov5s.yaml: Define la arquitectura del modelo basado en YOLOv5s (una versión pequeña de YOLOv5).
--weights yolov5s.pt: Se usaron los pesos preentrenados de YOLOv5 como punto de partida.
--name fruits: El modelo se guarda bajo el nombre fruits.

3. Validación y resultado del entrenamiento

Una vez organizado, configurado y las rutas del entrenamiento, se ejecuta el parámetro *python train.py --img 640 --batch 16 --epochs 100 --data C:\Users\SABTIZ\grado\capture\frutas.yaml --cfg models\yolov5s.yaml --weights yolov5s.pt --name fruits* desde el command promt de anaconda.

y empieza con el proceso de la siguiente manera en el repositorio:

- Finalizada la ejecución en el prompt como se muestra en la figura 10, se ve la salida del entrenamiento del modelo de detección de objetos usando YOLOv5

Progreso del entrenamiento:

- **“100 epochs completed in 29.721 hours.”** El entrenamiento completo de las 100 épocas tomó aproximadamente **29.7 horas**.

Optimización y validación:

- **"Optimizer stripped from..."**: Aquí se muestra que el optimizador ha sido removido del modelo final para reducir el tamaño de los archivos guardados:
  - **14.4 MB** es el tamaño del modelo guardado como best.pt y last.pt.
- **"Validating runs/train/fruits6/weights/best.pt..."**: Se realiza la validación del modelo entrenado para evaluar su rendimiento.

Resumen del modelo:

- **"157 layers, 7023610 parameters..."**: El modelo tiene:
  - **157 capas** en total.

- **7,023,610 parámetros** (valores que el modelo ajusta durante el entrenamiento).
- **15.8 GFLOPs** (operaciones en punto flotante, una métrica de la complejidad computacional).

- Valida los archivos de entrenamiento, por ejemplo encuentra las 1200 imágenes y las 300 imágenes para la validación.

```
python train.py --img 640 --batch 16 --epochs 100 --data C:\Users\SABTIZ\grado\capture\frutas.yaml --cfg
models\yolov5s.yaml --weights yolov5s.pt --name fruits

100 epochs completed in 29.721 hours.
Optimizer stripped from runs\train\fruits6\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\fruits6\weights\best.pt, 14.4MB

Validating runs\train\fruits6\weights\best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|
10/10 [00:53<00:00,  5.37s/it]
                 all        300        573      0.991      0.993      0.995      0.954
              lemon         300        101      0.997       0.99      0.995      0.944
          tangerine         300        120       0.99          1      0.995      0.983
             orange         300         97      0.997          1      0.995      0.989
             banana         300        142      0.986      0.973      0.993      0.865
               lulo         300        113      0.988          1      0.995      0.988
Results saved to runs\train\fruits6
```

Figura 10. Formato YOLOv5. *Fuente: Autores*

Resultados por clase:

La tabla número uno muestra la métrica para cada clase de fruta

CLASE	IMÁGENES	INSTANCIAS	P	R	mAP50	mAP50-95
all	300	573	0.991	0.993	0.995	0.954
lemon	300	101	0.997	0.990	0.995	0.944
tangerine	300	97	0.995	0.990	0.995	0.983
orange	300	97	0.997	1.000	0.995	0.989
banana	300	142	0.986	0.973	0.993	0.865
lulo	300	113	0.988	1.000	0.995	0.988

Tabla 1. Resultados por clase. *Fuente: Autores*

Cada métrica se define de la siguiente manera:

- **P (Precisión):** Proporción de predicciones correctas sobre el total de predicciones hechas.
- **R (Recall):** Proporción de objetos detectados sobre el total de objetos presentes.
- **mAP50:** Promedio de precisión para un umbral IoU del 50%.
- **mAP50-95:** Promedio de precisión en varios umbrales de IoU (del 50% al 95%).

Nota: Tengamos en cuenta que IoU significa intersección sobre unión. Es una métrica fundamental para evaluar el rendimiento de modelos de dirección y segmentación de objetos. Mide también las predicciones del modelo con los objetos reales de una imágen,

Ubicación de resultados:

- **"Results saved to runs/train/fruits6"**

  Los resultados (pesos entrenados y gráficas de entrenamiento) se guardaron en la carpeta runs/train/fruits6.

Validación de los resultados

En términos de reconocimiento, el modelo mostró un desempeño variado según el tipo de fruta y las condiciones en las que se tomaron las fotos. Los **limones** lograron un 100% de precisión, mientras que los **bananos** tuvieron un 80% de reconocimiento, pero solo cuando estaban cerca de la cámara, estos dos siendo los dos frutos para validación del sistema.

Para el entrenamiento del sistema se hicieron pruebas con otros tres frutos únicamente para analizar el funcionamiento de procesamiento y tenerlo en cuenta para adaptaciones futuras, así bien, los **lulos** obtuvieron un 60% de efectividad, y las **naranjas** alcanzaron apenas un 30%, pero solo bajo condiciones específicas de luz y a cierta distancia. Las **mandarinas** presentaron un reconocimiento limitado al 10%, pero únicamente cuando se sostenían en la mano, en un ángulo preciso.

El mejor desempeño se obtuvo con los limones, ya que las fotos incluyeron múltiples ejemplares a diferentes distancias, con diversas condiciones de luz y fondos. Sin embargo, se observó que sostener las frutas con la mano influye en el resultado, dado que el modelo detecta objetos por zonas de color. En estos casos, la piel de la mano puede interferir en la identificación al ser reconocida como parte

del fruto. Por este motivo, las mandarinas solo lograron ser correctamente detectadas cuando se sostenían en la mano bajo un ángulo específico.

Desarrollo de la interfaz gráfica

Finalmente, se creó una interfaz gráfica amigable que facilita la interacción con el software, permitiendo a los usuarios loguearse como se muestra en la figura 11 Y visualizar los resultados del conteo en tiempo real.

La estructura de la interfaz gráfica está basada en lógica de página web, es decir, se compone de:
Archivos estáticos CSS (archivos que no cambian y se consumen) funcionando como como hojas de estilo.
Imágenes sólidas que representan elementos.
Archivos JS que ejecutan funciones internas para el dinamismo de la página.

Siguiendo la siguiente estructura en resumen:

1. Archivo principal de estilos
2. Archivo de JS por cada página de visualización
3. Ejecución de un framework preinstalado FLASK

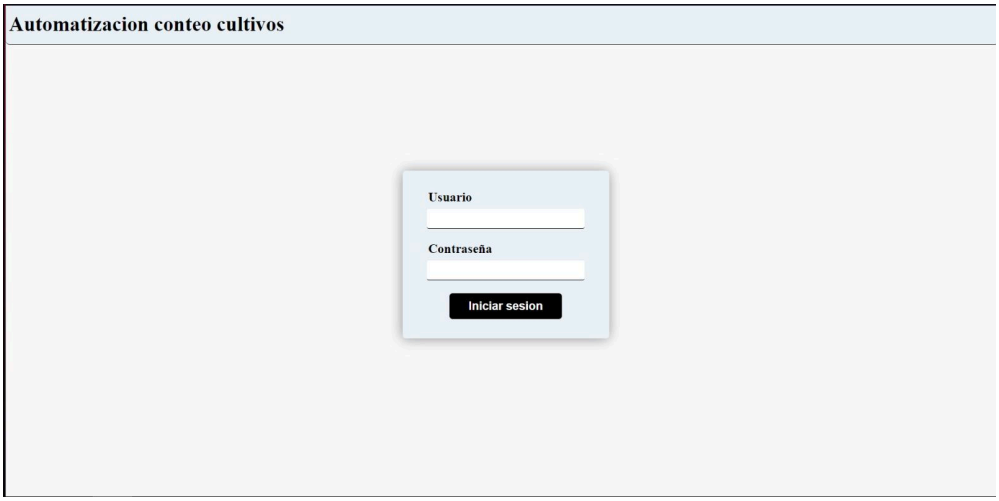


Figura 11. Interfaz ventana principal. *Fuente: Autores*

7.2. Restricciones de la solución

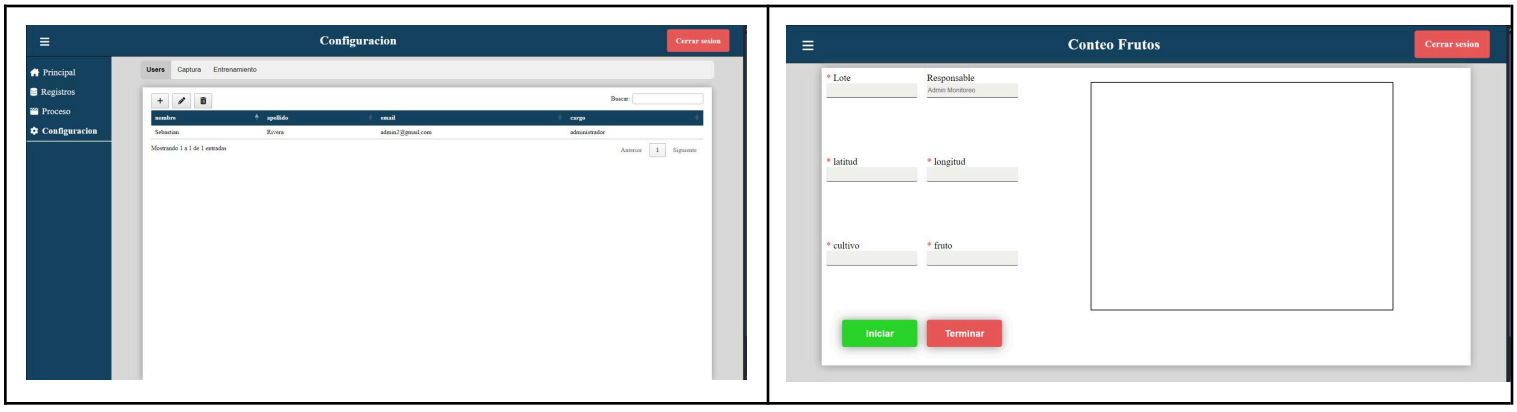
El proyecto fue diseñado desde su inicio teniendo en cuenta varias restricciones que son esenciales para garantizar su correcto desempeño. Las **condiciones de iluminación** deben ser consideradas cuidadosamente, ya que cambios en la luz, sombras o poca luminosidad pueden afectar la precisión del modelo. Asimismo, la

posición y el ángulo de captura son factores críticos, ya que se requiere que los frutos se capturen desde ángulos adecuados y estén completamente visibles para asegurar un reconocimiento fiable. Del mismo modo, la **distancia de la cámara** juega un papel fundamental, ya que el sistema fue diseñado para operar dentro de un rango específico, y fuera de este, su desempeño puede disminuir. Además, se contempla la **variabilidad natural de los frutos** en términos de tamaño, color o estado de maduración, lo que exige un modelo capaz de adaptarse incluso si las características de los frutos difieren de las incluidas en el conjunto de entrenamiento.

Otro aspecto clave considerado desde el diseño del proyecto es la **capacidad de procesamiento** del sistema, que puede requerir hardware con potencia suficiente para realizar inferencias en tiempo real, lo cual representa un desafío en entornos de campo o dispositivos móviles. Finalmente, se previó que el **entrenamiento y ajuste del modelo** demandaría tiempo y esfuerzo significativo, especialmente para asegurar que el sistema generalice correctamente en múltiples tipos de frutos y en diversas condiciones ambientales. Por tanto, todas estas restricciones fueron asumidas como parte del planteamiento inicial, con el objetivo de desarrollar una solución robusta y eficiente bajo las condiciones previstas.

7.3. validación de la solución (Resultados)

La interfaz cuenta con un menú en el cual se puede acceder a la pantalla principal, registros proceso y configuración como se muestra en la tabla 2. también donde se visualiza la imagen en tiempo real y se realiza el conteo. Cuenta con requerimientos como completar la información del fruto a contar



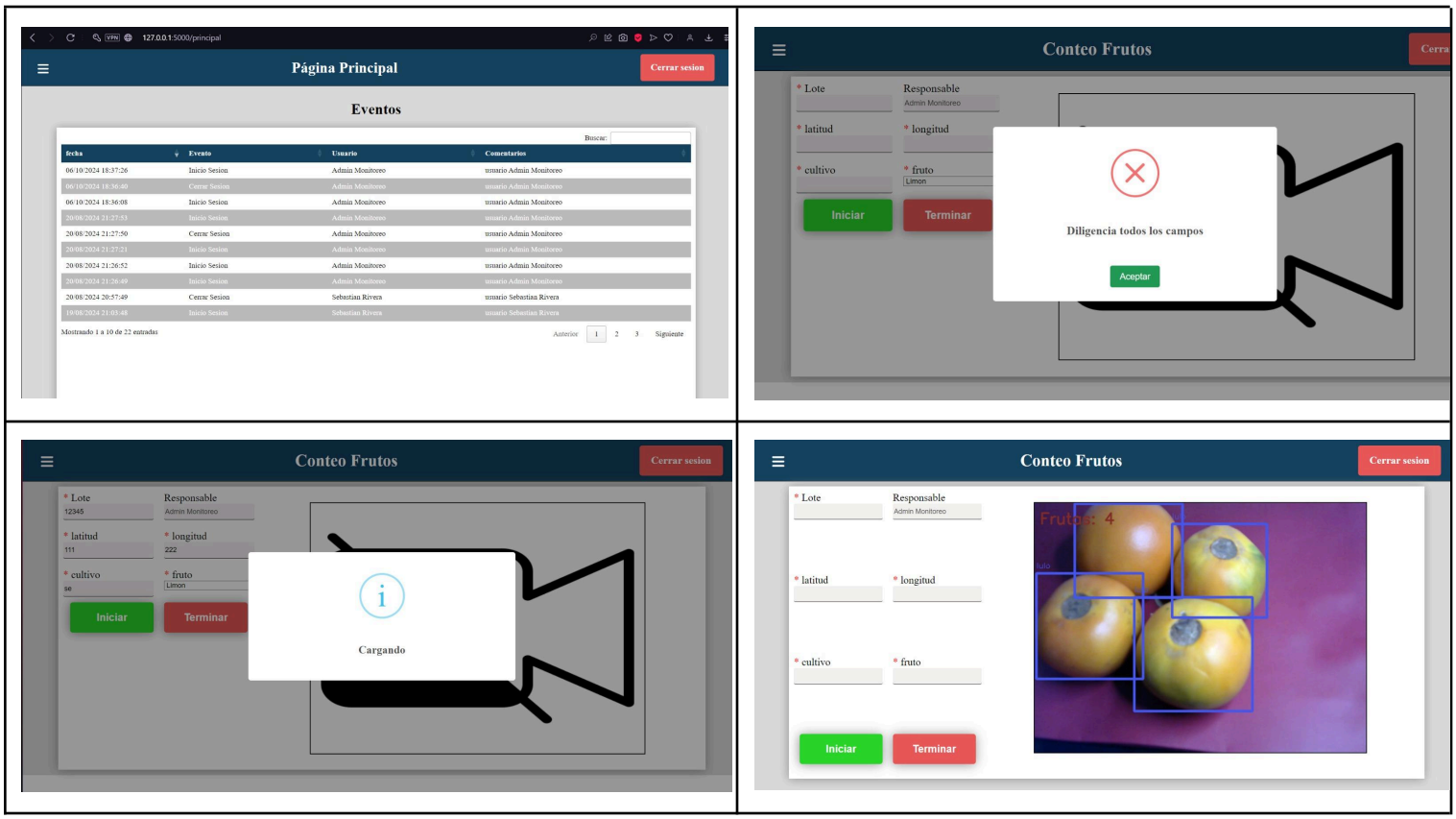


Tabla 2. Interfaz gráfica. *Fuente: Autores*

Cuando el modelo finaliza el entrenamiento nos arroja un directorio con los resultados, así se obtuvieron estadísticas de trenes y valores, mosaicos, etiquetas, predicciones y mosaicos aumentados, así como métricas y gráficos que incluyen curvas de recuperación de precisión y matrices de confusión.

Algunos de los resultados del entrenamiento y validación se muestran en las figuras 12 y 13. A simple vista se ve el conjunto de imágenes de frutos identificados automáticamente por el sistema, en la primera imágen se visualiza el entrenamiento y en la segunda es la validación, por eso los resultados de identificación son iguales.



Figura 12. Entrenamiento *Fuente: Resultados YOLOv5*



Figura 13. Validación *Fuente: Resultados YOLOv5*

En las **métricas de entrenamiento y validación** como se muestra en la figura 14 encontramos distintos gráficos que arrojan lo siguiente:

1. Losses (Pérdidas):

train/box_loss y **val/box_loss**: Estas pérdidas se refieren a qué tan bien se ajusta el modelo a las coordenadas de las cajas delimitadoras (bounding boxes). En ambos casos, las pérdidas disminuyen con las épocas, indicando que el modelo mejora su capacidad para ajustar las cajas a los frutos detectados.
train/obj_loss y **val/obj_loss**: Estas métricas reflejan la pérdida al detectar si existe un objeto en una región dada. La caída progresiva muestra que el modelo se vuelve más preciso para identificar la presencia de objetos.
train/cls_loss y **val/cls_loss**: Estas son las pérdidas de clasificación, es decir, qué tan bien el modelo clasifica los objetos detectados en las clases correspondientes. La reducción indica que el modelo está aprendiendo a clasificar correctamente los frutos con el tiempo.

2. Precisión y Recall:
   La **precisión** mide la proporción de predicciones correctas entre todas las predicciones hechas. En este caso, la precisión converge hacia 1, lo que sugiere que el modelo comete muy pocos falsos positivos.
   El **recall** refleja la proporción de objetos detectados correctamente entre todos los objetos presentes. El valor alto indica que el modelo está identificando la mayoría de los frutos, aunque al inicio del entrenamiento la curva tiene algunas fluctuaciones.

3. mAP (Mean Average Precision):
   mAP@0.5 muestra la precisión promedio para una IoU de 0.5 (es decir, 50% de coincidencia entre las cajas predichas y las reales).
   mAP@0.5:0.95 es una métrica más estricta, ya que promedia la precisión para IoUs desde 0.5 hasta 0.95. La convergencia hacia valores altos (cercanos a 0.8) indica que el modelo tiene un buen rendimiento en distintas condiciones.

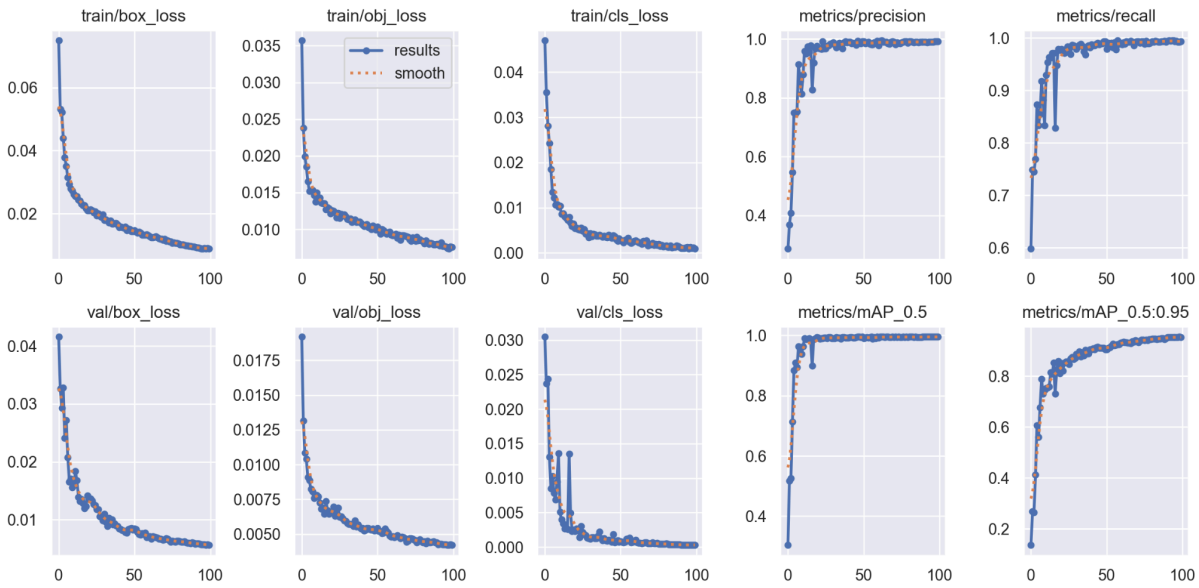


Figura 14. Results del entrenamiento y validación en métricas. *Fuente: Resultados YOLOv5*

Por otro lado en la figura 15 contamos con la **matriz de confusión** esta refleja cómo el modelo clasifica cada tipo de fruto en las clases disponibles. Los valores diagonales indican las predicciones correctas, mientras que los valores fuera de la diagonal representan los errores.

Como principales resultados de esta matriz se obtuvo:

Limón, mandarina, naranja y lulo: El modelo tiene un desempeño perfecto o casi perfecto para estas clases, con valores de 0.99 o 1.0 en las predicciones correctas.
Banana: El modelo logra un 98% de precisión, aunque en algunos casos se confunde y clasifica erróneamente un 1% como limón.
Fondo: En este caso, el modelo presenta algunas confusiones al detectar fondo como frutos, especialmente con lulo y naranja (40%).

Recordemos sin embargo que este sistema está diseñado para trabajar con los resultados el entrenamiento y la validación de los limones y las bananas, lo que no da excelentes resultados en la práctica real.

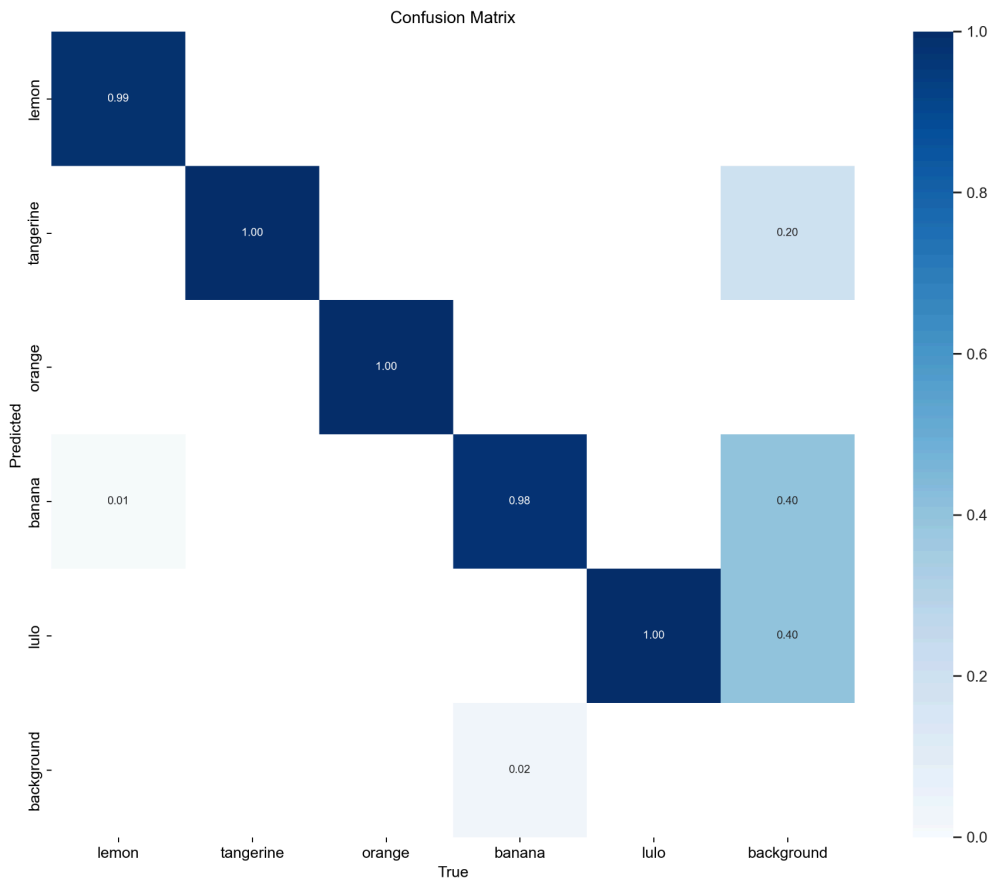


Figura 15. Matriz de confusión. *Fuente: Resultados YOLOv5*

En general, los resultados muestran que el modelo ha logrado un desempeño sólido, con pérdidas reducidas y métricas de precisión y recall elevadas. La mayor parte de las clases son correctamente identificadas con alta precisión, aunque hay ciertas confusiones menores, especialmente al distinguir el fondo de los frutos.

Al momento de hacer las pruebas se verificó el reconocimiento de los frutos, limones y bananos, cumpliendo el conteo y el reconocimiento como lo visualizamos en 16, se muestra el conteo de las 4 bananas y nos identifica el fruto, igualmente con los limones haciendo el conteo de 3 e identificando.

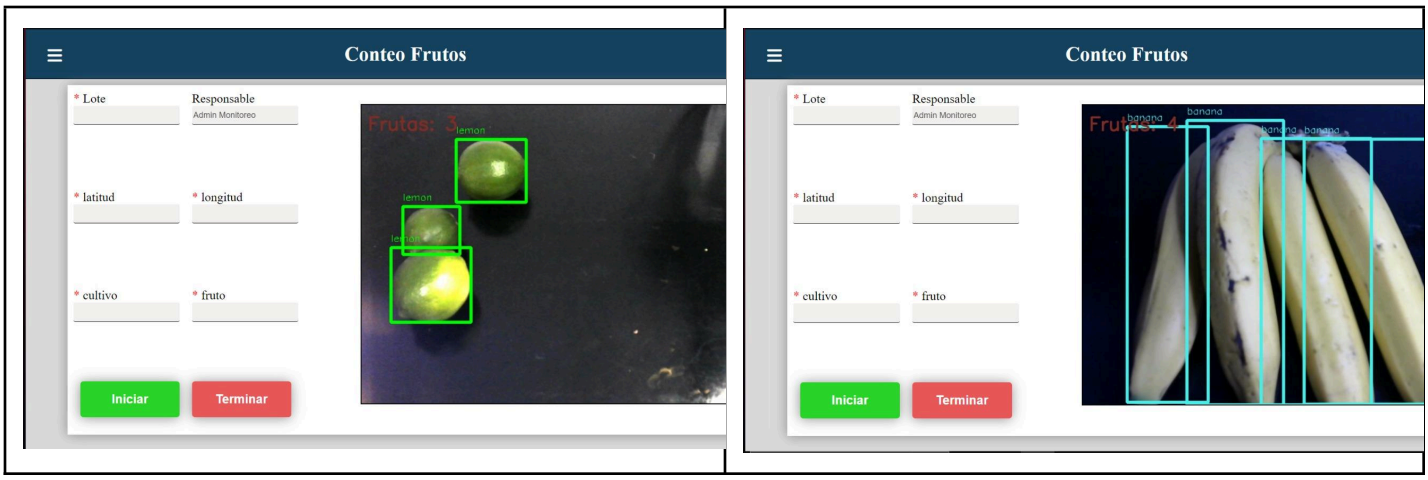


Figura 16. Resultados del conteo y detección de frutos. *Fuente: software diseñado para conteo y detección de frutos*

7.4. Evaluación del impacto de la solución

La automatización del conteo de frutos reduce los costos operativos al minimizar la dependencia de mano de obra para tareas repetitivas. Esto permite que los recursos humanos se asignen a labores más estratégicas y complejas. Además, el aumento en la precisión del modelo facilita una mejor gestión del inventario y optimiza las decisiones en los procesos de recolección, empaque y distribución. Esta mejora es especialmente relevante para pequeñas y medianas explotaciones agrícolas, donde el ahorro en tiempo y recursos tiene un impacto directo en la rentabilidad.

El proyecto demuestra la viabilidad del uso de redes neuronales como YOLOv5 en aplicaciones agrícolas, lo que abre las puertas a futuras innovaciones en este sector. La posibilidad de integrar visión artificial en dispositivos móviles amplía el alcance de la agricultura de precisión, permitiendo su implementación incluso en contextos rurales. Además, este tipo de solución tecnológica podría inspirar desarrollos más avanzados, como la detección automática de enfermedades en cultivos o la clasificación de frutos por calidad.

La introducción de tecnología avanzada en el sector agrícola tiene implicaciones positivas para la fuerza laboral. Al liberar a los trabajadores de tareas manuales repetitivas, estos pueden concentrarse en labores de mayor valor y seguridad, mejorando su calidad de vida. Adicionalmente, la implementación de sistemas de inteligencia artificial fomenta la capacitación y la especialización, preparando a los trabajadores para manejar nuevas herramientas digitales. Por otro lado, un conteo más preciso facilita la transparencia en los acuerdos comerciales entre productores y distribuidores, generando relaciones más justas y eficientes.

8. Conclusiones

A lo largo del desarrollo de este proyecto se logró diseñar e implementar un sistema basado en **redes neuronales profundas (YOLOv5)**, enfocado en la identificación y conteo automatizado de diferentes tipos de frutos. Los resultados obtenidos demuestran que el modelo es capaz de lograr un **alto rendimiento**, aunque también se identificaron limitaciones y aspectos a mejorar para su uso en entornos reales.

En cuanto al modelo mostró un **alto nivel de precisión** en la identificación de los frutos, como limón, y banano, con métricas cercanas al 100%. Sin embargo, en algunos casos, la clasificación del fondo generó confusiones con los frutos de entrenamiento, especialmente con lulo y naranja.

Las métricas de **mAP** (Average Precision) indicaron un desempeño robusto, alcanzando valores altos tanto para IoU de 0.5 como para el rango más estricto de 0.5 a 0.95. Esto confirma que el modelo es capaz de realizar detecciones precisas incluso bajo distintas condiciones de coincidencia.

Para el desempeño óptimo del modelo en cuanto a condiciones físicas se puede concluir que las pruebas revelaron que la variabilidad en las condiciones de luz tiene un impacto significativo en la precisión del modelo. Se comprobó que cambios drásticos de iluminación o sombras pueden afectar el reconocimiento.

El ángulo desde el cual se toma la imagen es fundamental. Se tiene que cumplir un criterio similar al del momento de la captura de las imágenes.

En cuanto a la distancia de la cámara en el modelo se observó dentro de un rango específico de distancia. Detecciones cercanas fueron más precisas (como en el caso de las bananas), mientras que distancias mayores presentaron más errores.

Se comprobó que la variabilidad en el tamaño, color o estado de maduración puede afectar el reconocimiento si estas variaciones no se reflejan en el conjunto de entrenamiento. En el caso de los limones, la alta precisión se logró porque el conjunto de datos incluía una amplia diversidad de ejemplos en diferentes condiciones y fondos.

El proceso de entrenamiento fue largo, con **100 épocas completadas en 29.7 horas**, lo que refleja la complejidad del modelo. Sin embargo, la curva de pérdidas muestra que el modelo converge adecuadamente, logrando optimizar la detección y clasificación durante las iteraciones.

En resumen, el proyecto logró cumplir con los objetivos planteados, proporcionando un sistema capaz de automatizar el conteo de frutos con alta precisión. Aunque se identificaron limitaciones, estas fueron consideradas desde el planteamiento inicial del proyecto, permitiendo adaptar las expectativas y optimizar el modelo dentro de un entorno controlado. Los resultados obtenidos son un punto de partida sólido para la implementación del sistema en aplicaciones prácticas, y las mejoras sugeridas podrían aumentar su versatilidad y robustez en escenarios más diversos.

9. Referencias

[1] Eissa, Mostafa. (2024). Precision Agriculture using Artificial Intelligence and Robotics. Journal of Research in Agriculture and Food Sciences.

[2] Bazargani, Khadijeh & Deemyad, Taher. (2024). Automation's Impact on Agriculture: Opportunities, Challenges, and Economic Effects. Robotics.

[3] Mejía Martínez, C. E., Ramirez Beltran, N. J., & Rojas Eslava, J. P. (2022). Software para automatización de conteo de plantas en cultivos de papa mediante visión artificial. Pontificia Universidad Javeriana https://repository.javeriana.edu.co/handle/10554/63696

[4] Martínez, C., Albornoz, E., Pizarro, J., & Carolo, y. R. (s/f). Sistema de Conteo Forestal, Agrícola y Ganadero mediante procesamiento de imágenes. Org.ar. Recuperado el 21 de octubre de 2024, de https://49jaiio.sadio.org.ar/pdfs/jui/JUI-10.pdf

[5] Universidad Militar Nueva Granada, Pulido-Rojas, C. A., Molina-Villa, M. A., Universidad Militar Nueva Granada, Solaque-Guzmán, L. E., & Universidad Militar Nueva Granada. (2016). Machine vision system for weed detection using image filtering in vegetables crops. Revista Facultad de Ingeniería Universidad de Antioquia, 80, 124–130.

[6] Pinho, Tatiana M.; Coelho, J.P.; Oliveira, Josenalde; Boaventura-Cunha, José (2017). Comparative analysis between LDR and HDR images for automatic fruit recognition and counting. Journal of Sensors. ISSN 1687-725X. p. 1-12

[7] Montoya Holguin, Christian, Cortés Osorio, Jimmy Alexander, & Chaves Osorio, José Andrés. (2014). Sistema automático de reconocimiento de frutas

basado en visión por computador. Ingeniare. Revista chilena de ingeniería, 22(4), 504-516. https://dx.doi.org/10.4067/S0718-33052014000400006

[8] Medina Tobón, D. (2021). Conteo de flores y frutos para el monitoreo del cultivo de aguacate Hass por medio de imágenes utilizando Machine Learning. Universidad de los Andes. Disponible en: http://hdl.handle.net/1992/52997

[9] Saleem, M.H., Potgieter, J. & Arif, K.M. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. Precision Agric 22, 2053–2091 (2021).

[10] Birrell, S., Hughes, J., Cai, J. Y., & Iida, F. (2020). A field‐tested robotic harvesting system for iceberg lettuce. Journal of Field Robotics, 37(2), 225–245.

[11] ¿Qué es la visión artificial? (2024, octubre 9). Ibm.com. https://www.ibm.com/mx-es/topics/computer-vision

[12] Qué es la Inteligencia Artificial. (s/f). Gob.es. Recuperado el 21 de octubre de 2024, de https://planderecuperacion.gob.es/noticias/que-es-inteligencia-artificial-ia-prtr

[13] ¿Qué es una base de datos? (2020, noviembre 24). Oracle.com; Oracle. https://www.oracle.com/co/database/what-is-database/

[14] (S/f). Amazon.com. Recuperado el 21 de octubre de 2024, de https://aws.amazon.com/es/what-is/neural-network

[15] de Imagina, E. (2024, octubre 21). Procesamiento de imágenes con OpenCV en Python. Imaginaformacion.com; Imagina Formación. https://imaginaformacion.com/tutoriales/opencv-en-python

[16] TensorFlow. (s/f). TensorFlow. Recuperado el 21 de octubre de 2024, de https://www.tensorflow.org/?hl=es-419

[17] Williams, J., & Thompson, R. (2019). *Labor challenges in modern agriculture: The role of automation in addressing workforce shortages*. Journal of Agricultural Economics and Development, 16(2), 45-58.

[18] Smith, A., & Robinson, K. (2020). *Automation in agriculture: Tackling labor shortages and improving efficiency*. International Journal of Agricultural Innovation and Research, 9(1), 22-34.

[19] C.M. Bishop. "Pattern recognition and machine learning". Springer. Vol. 1. New York, USA. 200

[20] N.B. Ahmad Mustafa, S. Khaleel Ahmed, Z. Ali, W. Bing Yit, A.A. Zainul Abidin and Z.A. Md Sharrif. "Agricultural Produce Sorting and Grading using Support Vector Machines and Fuzzy Logic". IEEE Conference Publications, pp. 391-396. 2009.

[21] R. González and R. Woods. "Digital Image Processing". Prentice-Hall. 3rd edition. 2007.