



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TELECOMUNICACIONES

Tema:

**SISTEMA INTELIGENTE DE SELECCIÓN DE COSECHA EN CULTIVO
DE PITAHAYA MEDIANTE VISIÓN ARTIFICIAL**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Telecomunicaciones

ÁREA: Comunicaciones

LÍNEA DE INVESTIGACIÓN: Tecnología de la Información y Sistemas de
control

AUTOR: Lenin Andrés Gutiérrez Sánchez

TUTOR: Ing. Ana Pamela Castro Martin, MSc.

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA INTELIGENTE DE SELECCIÓN DE COSECHA EN CULTIVO DE PITAHAYA MEDIANTE VISIÓN ARTIFICIAL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Lenin Andrés Gutiérrez Sánchez, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Ana Pamela Castro Martin, MSc.

TUTOR

AUTORÍA

El presente trabajo de titulación titulado: SISTEMA INTELIGENTE DE SELECCIÓN DE COSECHA EN CULTIVO DE PITAHAYA MEDIANTE VISIÓN ARTIFICIAL es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Lenin Andrés Gutiérrez Sánchez

C.C. 1804734349

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Lenin Andrés Gutiérrez Sánchez

C.C. 1804734349

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Lenin Andrés Gutiérrez Sánchez, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA INTELIGENTE DE SELECCIÓN DE COSECHA EN CULTIVO DE PITAHAYA MEDIANTE VISIÓN ARTIFICIAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Carlos Diego Gordón Gallegos, PhD
PROFESOR CALIFICADOR

Ing. Jesús Israel Guamán Molina, Mg
PROFESOR CALIFICADOR

DEDICATORIA

El presente trabajo es el fruto del esfuerzo y sacrificio de mi perseverancia bajo la conducción y protección directa de nuestro Padre Celestial, la Virgen María y mi niño Jesús quienes me otorgaron sabiduría, conocimiento e iluminación para concluir con éxito el objetivo propuesto al inicio de mi carrera profesional.

A la memoria de mi adorable madrecita Jaqueline que espiritualmente siempre me está cuidando y protegiendo con su amor maternal desde la morada sagrada que Dios lo preparó.

A mi padre Augusto, mis hermanos María Augusta y Carlos Daniel quienes me dan la fortaleza y virtud con sus consejos, enseñanzas y el ejemplo para poder alcanzar las metas trazadas por más obstáculos que se presenten en el diario vivir

¡Gracias familia por ser el pilar fundamental y el camino de mi carrera profesional! ¡Siempre los llevaré en mi corazón!

Lenin Gutiérrez

AGRADECIMIENTO

Empezaré agradeciendo a Dios, la Virgencita y el niño Jesús por ser el camino de la luz y el horizonte de las derrotas y triunfos en mi carrera profesional, a mis abuelitas Gladys (+) y Fidelia (+) por ser mis primeras madres que me dedicaron su tiempo para conducir por el camino del bien formándome como persona con principios y valores.

Un agradecimiento muy especial a mi hermana Ing. María Augusta por ser el motor dinamizador y llegar a obtener la preseña tan abnegada siendo el nuevo profesional de la familia.

No puedo pasar por alto a la familia Bravo Sánchez quienes me apoyaron incondicionalmente para culminar mi trabajo investigativo en el proceso de alcanzar mi meta profesional.

Agradezco a todos mis profesores de nuestra alma mater de manera especial a la Ing. Ana Pamela Castro Martin, MSc. quien me asesoró y guió en este proyecto, poniendo en práctica su cúmulo de experiencias con su conocimiento en el proceso para culminar con éxito mi carrera profesional.

Por último, mi agradecimiento fraterno e incondicional a todo el círculo de personas que estuvieron junto a mí, apoyándome en el proceso formativo hasta alcanzar el logro que se desea obtener cada una de las familias.

Lenin Gutiérrez

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xiv
ÍNDICE DE ANEXOS	xvii
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	3
1.3 Fundamentación teórica	6

1.3.1 Agricultura de precisión	6
1.3.2 Control de calidad de las frutas	9
1.3.3 Selección de la cosecha	12
1.3.4 Inteligencia artificial (IA)	14
1.3.5 Procesamiento de imágenes	15
1.3.6 Visión artificial.....	16
1.4 Objetivos	24
1.4.1 Objetivo general	24
1.4.2 Objetivos específicos	24
CAPÍTULO II.- METODOLOGÍA.....	26
2.1 Materiales.....	26
2.2 Métodos.....	26
2.2.1 Modalidad de la investigación	26
2.2.2 Recolección de información.....	27
2.2.3 Procesamiento y análisis de datos	27
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN	28
3.1 Análisis de la producción de pitahaya en el Ecuador.....	28

3.1.1 Producción de pitahaya en el Ecuador	28
3.1.2 Ubicación	29
3.1.3 Operaciones básicas de cosecha de la pitahaya.....	29
3.1.4 Exportación	30
3.2 Arquitectur del Sistema.....	31
3.2.1 Selección de hardware.....	32
3.2.2 Sistema de Iluminación	33
3.2.3 Selección de software.....	34
3.3 Diseño del Sistema.....	38
3.3.1 Entrenamiento del Sistema.....	38
3.3.2 Código de Python.....	46
3.3.3 Base de Datos	51
3.3.4 Modelo de la tabla.....	52
3.3.5 Interfaz Gráfica del Sistema de Control de Calidad de la Pitahaya	53
3.3.6 Diagrama de flujo del sistema.....	56
3.4 Pruebas del entrenamiento YOLOv5	59

3.4.1 Distribución de clases	59
3.4.2 Validación de resultados	60
3.4.3 Indicador F1	62
3.4.4 Imágenes del testeo de clasificación de la pitahaya	63
3.4.5 Pruebas de funcionamiento del sistema	64
3.5 Confiabilidad del prototipo	71
3.6 Presupuesto	71
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES.....	73
4.1 Conclusiones	73
4.2 Recomendaciones.....	74
REFERENCIAS BIBLIOGRÁFICAS	75
ANEXOS	84

ÍNDICE DE TABLAS

Tabla 1. Calibres de los frutos de acuerdo con la masa unitaria	10
Tabla 2. Calibres para exportación de pitahaya amarilla “Palora”. Nota: Toma de FRAPELIVE S.A.	11
Tabla 3. Categoría de la Pitahaya Amarilla	11
Tabla 4. Color de maduración de la pitahaya amarilla	13
Tabla 5. Etapas de un proceso de Visión Artificial.....	17
Tabla 6. Base de Datos	21
Tabla 7. Técnicas utilizadas en cultivos de frutas por visión artificial	22
Tabla 8. Recolección de información.....	27
Tabla 9. Calibres para exportación de pitahaya amarilla “Palora”	31
Tabla 10. Selección de cámara web	33
Tabla 11. Sistema de iluminación	33
Tabla 12. Software de Desarrollo.....	34
Tabla 13. Aplicación para código fuente	35
Tabla 14. Sistema de entrenamiento del modelo de visión artificial.	36
Tabla 15. Base de datos.....	37
Tabla 16. Resultado de 135 épocas.....	60
Tabla 17. Resultado de 200 épocas.....	61
Tabla 18. Pruebas de funcionamiento del sistema.	67
Tabla 19. Eficacia del sistema.....	70

Tabla 20. Presupuesto.	71
-----------------------------	----

ÍNDICE DE FIGURAS

Figura 1. Las tres etapas de la Agricultura de Precisión	7
Figura 2. Sensor en cultivo de pitahaya	8
Figura 3. Pitahaya Amarilla.	9
Figura 4. Pitahaya amarilla para consumo	10
Figura 5. Grados de maduración de la Pitahaya Amarilla	13
Figura 6. Ejemplo de red neuronal	15
Figura 7. Ejemplo de binarización de una imagen	16
Figura 8. Resultados de detección de varias frutas	16
Figura 9. Visión Artificial.....	17
Figura 10. Diagrama de bloques de las etapas de un sistema de visión artificial	18
Figura 11. Diagrama de bloques de un sistema SVA.....	19
Figura 12. Arquitectura de la red convolucional YOLO.....	20
Figura 13. Arquitectura Yolov5	21
Figura 14. Mapa de ubicación de la finca (Google Maps).	29
Figura 15. Proceso de cosecha de la pitahaya.	30
Figura 16. Grados de maduración de pitahaya.	30
Figura 17. Esquema General Sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial.	32
Figura 18. Imágenes de los grados de maduración de la pitahaya amarilla.	38
Figura 19. Proceso de adquisición de las imágenes para entrenamiento.	39

Figura 20. Interfaz, selección de imágenes y etiquetado ya sea por grados o etapas.	40
Figura 21. Selección de las imágenes de la pitahaya para cada grado de maduración	40
Figura 22. Labels generados.	41
Figura 23. Creación de carpetas para entrenamiento y validación.....	42
Figura 24. Labels de entrenamiento y validación.	42
Figura 25. Compresión de la carpeta data y subida a Google Drive.....	43
Figura 26. Importación de librerías y declaración de variables.	46
Figura 27. Conexión a YOLO.....	47
Figura 28. Función enviar.	47
Figura 29. Realizar una petición por URL.	47
Figura 30. Enviar el historial.....	48
Figura 31. Función mostrar el historial.	48
Figura 32. Función capturar.	49
Figura 33. Tratamiento de imágenes para determinar el porcentaje de color verde. .	50
Figura 34. Controlar la etapa1.....	50
Figura 35. Codificación para el tratamiento de la etapa3.....	51
Figura 36. Base de datos	51
Figura 37. Tabla de datos de selección de la pitahaya.	52
Figura 38. Registros en la base de datos.	52
Figura 39. Archivos PHP.	52

Figura 40. Interfaz Gráfica del Sistema de Control de Calidad de la Pitahaya.....	54
Figura 41 Detección actual de las frutas con una columna con el número de pitahayas y una segunda columna con el historial de las pitahayas según sus grados	55
Figura 42. Esquema de la adquisición de la imagen.	55
Figura 43. Cabina de visión artificial.	56
Figura 44. Implementación del prototipo.....	56
Figura 45. Diagrama de flujo del sistema.	58
Figura 46. Distribución de clases.	59
Figura 47. Matriz de confusión.	61
Figura 48. Indicador F1.....	63
Figura 49. Imágenes del testeo de clasificación de la pitahaya.....	63
Figura 50. Prototipo implementado del sistema de sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial	64
Figura 51. Pruebas de funcionamiento pitahaya etapa 1.....	65
Figura 52. Pruebas de funcionamiento pitahaya etapa 2.....	65
Figura 53. Pruebas de funcionamiento pitahaya etapa 3.....	66
Figura 54. Pruebas de funcionamiento del sistema	67

ÍNDICE DE ANEXOS

Anexo A. Entrevista realizada a jefe y empleados de la finca	84
Anexo B. Norma Técnica Ecuatoriana NTE INEN 2003:2005	85
Anexo C. Datasheet Cámara Web Trust Gaming GXT 1160 Vero.	86
Anexo D. Led Panel RD.....	87
Anexo E. Diseño de la Cabina de Visión Artificial.	88
Anexo F. Código de Python.	92
Anexo G. Posición de la cámara web en el sistema de visión artificial.....	102

RESUMEN EJECUTIVO

El control de calidad de las frutas de exportación en el Ecuador como la pitahaya amarilla es fundamental ya que esta tiene gran aceptación en mercado nacional e internacional. La norma técnica NTE INEN 025 establece criterios de clasificación de la fruta mediante los grados de maduración para exportación y consumo nacional. El presente trabajo investigativo tiene como objetivo el desarrollo de un sistema de visión artificial de selección de cosecha en cultivo de pitahaya que permite al agricultor una selección de esta fruta para su exportación de manera automática evitando el error humano. Este sistema implica la utilización del modelo YOLOv5 para el aprendizaje profundo en el contexto agrícola la cual evalúa la maduración de pitahayas en tres etapas.

El prototipo se basa en una cabina compuesta por elementos primordiales para la visión artificial que es una cámara web de alta resolución y luz blanca artificial. El sistema captura la imagen de seis frutas y clasifica según el grado de maduración de la pitahaya en un tiempo estimado de 4.48 segundos. Para el procesamiento de datos se utiliza una interfaz gráfica programada en Python.

Se llevaron a cabo pruebas para evaluar la funcionalidad y verificación del prototipo, obteniendo un nivel de confiabilidad del 91,82%. Estas pruebas involucraron diversas pitahayas en distintos grados de maduración, lo que permite concluir que el sistema resulta útil para agricultores o personas encargadas de clasificar la pitahaya amarilla destinada tanto a la exportación como al consumo nacional.

Palabras clave: YOLOv5, visión artificial, selección de cultivos, grados de madurez.

ABSTRACT

Quality control of export fruits in Ecuador, such as yellow pitahaya, is essential because it is widely accepted in both domestic and international markets. The technical standard NTE INEN 025 establishes criteria for the classification of fruit by degrees of ripeness for export and domestic consumption. The objective of this research work is the development of an artificial vision system for crop selection in pitahaya cultivation that allows the farmer to automatically select this fruit for export, avoiding human error. This system involves the use of the YOLOv5 model for deep learning in the agricultural context, which evaluates the ripening of pitahayas in three stages.

The prototype is based on a cabin composed of primary elements for artificial vision, which is a high-resolution webcam and artificial white light. The system captures the image of six fruits and classifies them according to the degree of ripeness of the pitahaya in an estimated time of 4.48 seconds. A graphical interface programmed in Python is used for data processing.

Tests were carried out to evaluate the functionality and verification of the prototype, obtaining a reliability level of 91.82%. These tests involved various pitahayas at different degrees of ripeness, which leads to the conclusion that the system is useful for farmers or persons in charge of grading yellow pitahayas destined for both export and domestic consumption.

Keywords: YOLOv5, artificial vision, crop selection, maturity grades.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA INTELIGENTE DE SELECCIÓN DE COSECHA EN CULTIVO DE PITAHAYA MEDIANTE VISIÓN ARTIFICIAL

1.1.1 Planteamiento del problema

En los últimos años se creó un método conocido como “agricultura de precisión” con la intención de aumentar la eficacia y la rentabilidad de la producción agrícola. Utilizando herramientas tecnológicas, se analizan las variaciones del suelo, el clima y otros factores que afectan el desarrollo de las plantas. Con este conocimiento, las opciones sobre el uso de fertilizantes, herbicidas y otros insumos pueden hacerse con mayor precisión, lo que permite ahorros financieros y limita el efecto ambiental de las operaciones agrícolas [1].

América contribuye con el 21.4 % de la producción mundial de frutas. De su producción total, el 25 % se destina a la exportación, lo que sugiere una baja demanda interna. Esta situación se agrava en los países subdesarrollados, donde las exportaciones representan más del 70 % de la producción. América es una de las principales regiones productoras y exportadoras de frutas en todo el mundo. En términos de producción, se sitúa en segundo lugar, solo detrás de Asia, siendo: Costa Rica, Ecuador y Estados Unidos sus principales proveedores lo que refleja la diversidad y el potencial que existe en la región para el cultivo y la exportación de frutas de alta calidad [2].

En 2021, la agricultura en América Latina representó alrededor del 13% de la producción total agrícola. Según las previsiones, la producción de frutas y hortalizas en Colombia podría crecer un 4,5% anual hasta 2025. Mediante las investigaciones realizadas, la producción agrícola crecerá alrededor de un 1,4% durante los próximos diez años. Según las proyecciones en la producción de frutas en América Latina, la misma seguirá creciendo durante los próximos años debido al aumento de la demanda del mercado internacional y la inversión en tecnología y sostenibilidad [3].

En el Ecuador más del 64% de la producción agrícola está en manos de pequeños productores. [4] Sin embargo, a pesar de los importantes avances tecnológicos, la producción agrícola en el Ecuador sigue siendo mínima. Hay varias razones para esta baja productividad, incluyendo la disponibilidad restringida de fertilizantes, inversión en tecnología y el escaso uso de los sistemas de riego. El uso restringido de la tecnología en la agricultura, es uno de los principales inconvenientes que no permite optimizar y mejorar los procesos agrícolas particularmente en el ámbito de la pequeña producción [5].

El enfoque actual de productores y consumidores es producir y consumir alimentos más seguros, por lo que el control de calidad se ha convertido en un proceso imprescindible para mantener más alto el nivel de calidad de las frutas. Todos los productores y distribuidores de alimentos están obligados a pasar el control de calidad, pero en el caso de las frutas y verduras su costo está cada vez más relacionado con la calidad del producto final [6].

La implementación de herramientas tecnológicas modernas facilitará una clasificación más eficiente de la pitahaya, permitiendo su selección tanto para la industrialización alimentaria como para la exportación a países extranjeros. Además, contribuirá al consumo local, mejorando así los ingresos económicos de los agricultores en el oriente ecuatoriano, con la provincia de Morona Santiago, específicamente el cantón Palora, destacándose como la principal productora de esta fruta. Las condiciones geográficas de este cantón permiten que las frutas sean más grandes, con mayor contenido de materia seca, peso, y de mejor apariencia física, con respecto a las otras provincias que también se cultivan [7].

La calidad y la vida útil de la fruta puede verse significativamente afectada por una infraestructura inadecuada para el almacenamiento y la entrega de producción cultivada. La fruta que no se conserva correctamente puede degradarse rápidamente y no llegar a su destino final en excelente condición, por lo que los productores y consumidores pueden sufrir pérdidas económicas. Por otro lado, si la fruta se transporta y se almacena de manera adecuada se prolongará la vida útil preservando la calidad de la fruta. Además, los clientes tendrán más confianza en la fruta que adquieren porque conocen que fue transportada y almacenada en condiciones

adecuadas, por tal razón es necesario realizar inversiones en la infraestructura moderna para conservar el producto fresco [8].

La eficacia y la calidad de los productos agrícolas pueden verse significativamente afectados por la falta de capacitación en métodos y técnicas de cosecha. Los agricultores que no conocen adecuadamente el proceso técnico-agrícola pueden cometer errores que causen pérdida del producto. También desconocen la tecnología y herramientas más recientes aplicables en la cosecha. Por otro lado, el personal debidamente capacitado y conocedor de los métodos de cosecha y poscosecha más recientes son eficientes y no cometen errores al cosechar y almacenar la fruta. Como consecuencia, los pequeños y grandes productores de cultivo de frutas obtendrán mayor rentabilidad económica ofertando productos de mejor calidad para el mercado interno y externo del país [9].

1.2 Antecedentes investigativos

En el presente trabajo se realizó una búsqueda de bibliográficas a nivel nacional e internacional. En estas fuentes se encontraron trabajos relacionados con el tema de investigación, los mismos que fueron utilizados como referentes y orientaciones para el desarrollo del presente proyecto a ejecutarse. Los estudios más relevantes se presentan en los siguientes párrafos:

En el año 2016, Herrera J., Medina S., Beleño K. y Gualdrón O. llevaron a cabo una investigación titulada "Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial" en la Universidad Autónoma de Caribe. En el prototipo, se emplea una técnica que implica el uso de una cinta transportadora para desplazar el café mientras el sistema de imágenes captura instantáneas de la fruta. Posteriormente, estas imágenes son sometidas a análisis mediante algoritmos de procesamiento de imágenes, encargados de extraer características de color y rasgos específicos del café. Los parámetros obtenidos son luego enviados a un algoritmo de clasificación y detección, que identifica la presencia de la plaga del café conocida como broca. Se realizó una validación del estado de los frutos de café utilizados como muestra mediante el método de flotación tradicional. Esto permitió la comparación

de los resultados obtenidos por los algoritmos con el estado real de los frutos. El sistema propuesto logró un rendimiento del 87% [10].

En el año 2019, Villota P., estudiante de la Universidad de Cuenca, realizó la investigación “Implementación de una estación prototipo con visión artificial, aplicado a la agricultura de precisión” La agricultura de precisión se divide en tres etapas, comenzando con el uso de técnicas de procesamiento de imágenes para detectar frutos en una parcela. Posteriormente, se aplican dos tecnologías basadas en el Internet de las Cosas (IoT). Específicamente, se utiliza la computación en la nube para gestionar la información generada por los sensores, y el protocolo de transporte de mensajería de cola (MQTT) se emplea para la transmisión de datos. Los resultados obtenidos mediante los algoritmos GMM con EM muestran una segmentación mejorada, aunque con un rendimiento más lento en comparación con el algoritmo de umbralización. Esto se debe a una menor incidencia de falsos positivos. Sin embargo, los costos de procesamiento son notablemente mayores, requiriendo aproximadamente 21 minutos para procesar una imagen en la Raspberry Pi. Además, se confirmó la eficacia del protocolo MQTT al transmitir datos y vídeo, con tiempos de latencia aceptables para la aplicación diseñada [11].

La IEEE de Italia, en el año 2020 realizó la siguiente investigación “Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application”, por Mazzia V., Khaliq A. y Chiaberge M., El sistema utiliza el algoritmo YOLOv3-tiny, una red diseñada para la detección rápida y precisa de objetos en tiempo real, inspirada en "Edge AI". Se implementa para la detección de manzanas en diversas plataformas integradas, como Raspberry Pi 3 B+, en combinación con Intel Movidius Neural Computing Stick (NCS), Nvidia Jetson Nano y Jetson AGX Xavier. La arquitectura YOLOv3-tiny demuestra su capacidad para detectar objetos pequeños, mostrando la viabilidad de implementar el modelo personalizado en hardware integrado de bajo costo y consumo. Esto se logra sin comprometer la precisión media de detección, alcanzando un 83,64%, y logrando frecuencias de imagen de hasta 30 fps, incluso en escenarios desafiantes, como la detección de manzanas superpuestas [12].

En China Agricultural University, Nazrul I. y Owais A., llevaron a cabo la investigación titulada "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques" en el año 2022. Al implementar un sistema compuesto por tres elementos: 1) un clasificador de frutas basado en visión por ordenador utilizando aprendizaje profundo, 2) un módulo de cámara Raspberry Pi para la captura y verificación de imágenes en tiempo real, y 3) una pantalla táctil frontal con una interfaz gráfica de usuario para mostrar los resultados de la clasificación en tiempo real, el prototipo del sistema de hardware consta de dos módulos Raspberry Pi. Uno de ellos es el kit de visión Google AIY, que incluye una Raspberry Pi Zero acelerada por visión para la inferencia de redes neuronales y una cámara Pi. El otro módulo es una Raspberry Pi con una pantalla táctil para mostrar información en tiempo real. Después de someter al sistema a entrenamiento y pruebas con dos conjuntos de datos, específicamente manzanas y plátanos, se obtuvo una precisión media del 99,2% y 98,6%, respectivamente, al utilizar el modelo EfficientNet en los conjuntos de prueba correspondientes. Además, la implementación de métodos de aprendizaje profundo a partir del conjunto de apilamiento resultó en una mejora ligeramente notable en la tasa de reconocimiento, con un aumento del 0,03% para las manzanas y del 0,06% para los plátanos [13].

En la investigación realizada por Vivar W. y Vivas M. en el año 2022 en la Universidad Politécnica Salesiana Sede Guayaquil, se llevó a cabo la investigación "Diseño e Implementación de un Sistema de Control Automático con Visión Artificial y Redes Neuronales destinado al Control de Calidad de Alimentos", la base técnica del proyecto se enfoca en la clasificación y selección de frutas mediante un sistema de inteligencia artificial implementado en una Raspberry Pi. La red neuronal preentrenada, ejecutada en este dispositivo, realiza una visión en tiempo real de los objetos, clasificándolos según sus características y asegurando la producción de frutas de alta calidad. La parte mecánica del proyecto está a cargo de dos Arduinos que controlan el movimiento de las frutas a través de un motor conectado a la cinta transportadora. La comunicación con la red neuronal facilita la clasificación mediante servomotores y pistones. Además, el sistema cuenta con sensores de proximidad para detectar las posiciones de los objetos en la cinta transportadora. El prototipo incluye una pantalla para monitorear el proceso en tiempo real, lo que contribuye a reducir

posibles errores. Con la implementación de tecnología e infraestructura, este sistema optimiza la clasificación de frutas. La operación de la red se beneficia del reconocimiento visual de la fruta, identificando manzanas en mal estado con una precisión del 0,61% y una velocidad de fotogramas de 4,1 FPS. Esto asegura que tanto las industrias como los consumidores tengan acceso a productos de alta calidad y procesos más eficientes [14].

En el año 2022, en la Universidad Técnica de Ambato, el ingeniero Garcés A. llevó a cabo una investigación sobre un "Sistema cuantificador de calidad de cultivo de manzana para monitoreo de la producción utilizando algoritmos de Aprendizaje Profundo con Visión Artificial y Segmentación de Instancias". Se logró identificar las variedades de manzanas mediante el modelo SSD-MobileNet, mientras que, para evaluar la calidad en distintos niveles, se empleó una red convolucional FCNResNet-18. Este sistema proporciona dos análisis: la identificación de las variedades de manzanas y la medición de su calidad, permitiendo la inspección y validación de manera no invasiva. La evaluación del rendimiento del sistema de segmentación y cuantificación de la calidad de las manzanas se basó en la precisión del entrenamiento, arrojando una media de IoU (Intersección sobre Unión) del 54,65% y una precisión del 94,56% [15].

1.3 Fundamentación teórica

Con el fin de desarrollar de manera precisa y enfocada el sistema, se ha llevado a cabo la recopilación y análisis de información proveniente de investigaciones que tienen una estrecha conexión con el tema planteado.

1.3.1 Agricultura de precisión

La agricultura de precisión abarca diversas estrategias diseñadas para optimizar el aprovechamiento de recursos agrícolas, como semillas, productos químicos y enmiendas. Su enfoque se centra en medir la variabilidad tanto en el espacio como en el tiempo de la producción agrícola. La clave radica en aplicar la cantidad justa de estos insumos, considerando las particularidades y demandas de cada ubicación específica dentro de las áreas cultivadas [16].

La agricultura de precisión no solo implica analizar la variabilidad en ubicaciones específicas, sino también implementar tácticas de gestión que consideren esta diversidad. Por lo general, aquellos que desarrollan y adoptan los procedimientos de la agricultura de precisión suelen categorizar estas tecnologías en tres fases distintas. En la Figura 1, se presentan las etapas de la agricultura de precisión. 1) Recolección de datos; 2) Procesamiento e interpretación de la información, y 3) Aplicación de insumos [16].

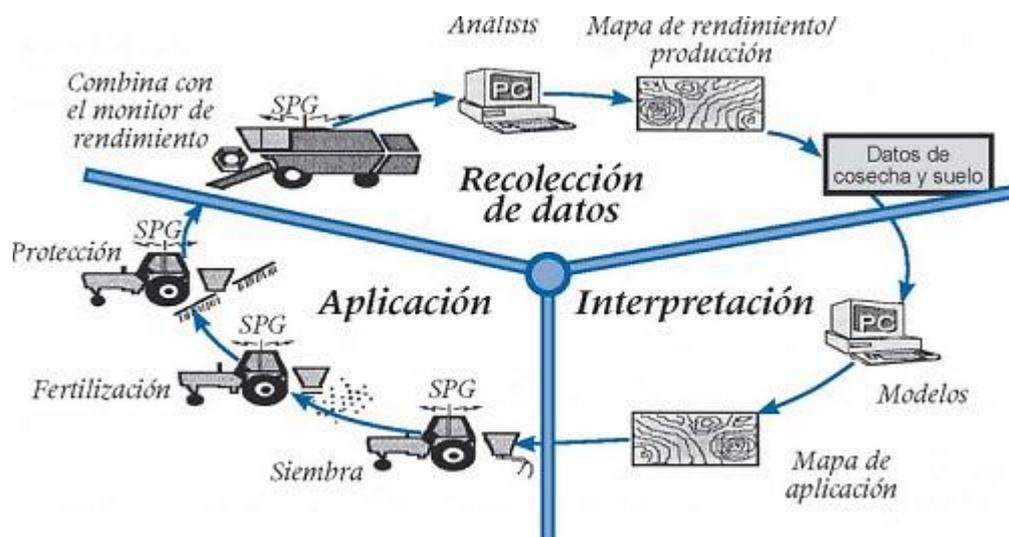


Figura 1. Las tres etapas de la Agricultura de Precisión [16].

a. Herramientas que se usa en la agricultura de precisión

Los agricultores pueden acceder a estas tecnologías de agricultura de precisión a nivel mundial. [16]:

- Cosechadoras equipadas con monitores de rendimiento y tecnología GPS.
- Sembradoras que incorporan un sistema electrohidráulico para ajustar de manera variable la dosificación de dos fertilizantes y la densidad de siembra.
- Pulverizadores equipados con navegación por satélite, piloto automático y capacidad para aplicar dosis variables de herbicidas.
- Aviones de señalización guiados por satélite.

- Equilibradoras/correctoras de pH equipadas con un sistema electrohidráulico que permite una dosificación ajustable.
- Dispositivos de teledetección: capturas satelitales, fotografías tomadas desde el aire y sensores en la banda del espectro cercano.
- Muestreo de suelos basado en áreas de gestión.
- Plataformas de Geoinformación.

b. Agricultura de precisión en la producción de fruta

La agricultura digital, gracias a una variedad de herramientas técnicas, posibilita que el sector agrícola supervise distintas variables del suelo y del entorno, recopilando y analizando datos de manera eficiente [17].

Este enfoque técnico emplea diversos sensores para recolectar información acerca de los factores que inciden en los cultivos. Posteriormente, los datos recopilados se envían a una plataforma web intuitiva, brindando a los agricultores la capacidad de interpretar de manera precisa la información y tomar decisiones informadas. Al adoptar este concepto, los agricultores pueden acceder a asesoramiento y respaldo a lo largo de todas las etapas del ciclo de producción agrícola [17]. Como se muestra en la Figura 2, la utilización de tecnología (sensores) aplicada en cultivos de pitahaya.



Figura 2. Sensor en cultivo de pitahaya [17].

1.3.2 Control de calidad de las frutas

La inspección de las propiedades físicas y químicas de las frutas y hortalizas es una parte integral del procedimiento de control de calidad, con el propósito de garantizar beneficios tanto para los alimentos como para los consumidores. Esta actividad se desglosa en dos etapas: la evaluación sensorial y la evaluación técnica [18].

Los sentidos desempeñan un papel fundamental en la valoración de la fruta, considerando aspectos como su aroma, tonalidad, consistencia, sabor y apariencia, incluyendo la uniformidad en tamaño y color. Estos atributos se evalúan mediante un análisis sensorial [18].

Además, el análisis físico-químico constituye una parte esencial de la evaluación técnica, examinando la composición y las condiciones microbiológicas de la fruta o verdura mediante el empleo de instrumentos y equipos de laboratorio. Este análisis aborda aspectos como la acidez, el índice de refracción, el contenido de sólidos solubles, la materia seca, la humedad y las cenizas, el ácido ascórbico, las vitaminas, las proteínas, la densidad y el pH de la fruta o verdura, entre otros [18].

a. Pitahaya

La pitahaya, conocida popularmente como la “fruta del dragón”, es un fruto exótico originario de Centroamérica y países de Sudamérica. Clasificada científicamente como *Selenicereus megalanthus*, como se muestra en la Figura 3 pertenece a la familia de los cactus. Valorada por su pulpa jugosa y refrescante, así como por su piel escamosa y sus llamativas tonalidades que abarcan desde el rosa hasta el amarillo [19].



Figura 3. Pitahaya Amarilla.

b. Normas del control de calidad de la pitahaya en el Ecuador

Las pitahayas deben cumplir con ciertos requisitos físicos mínimos, que incluyen mantenerse intactas y sin lesiones, presentar la forma ovoide característica de la fruta, mostrar frescura y una consistencia firme. El tallo debe medir entre 15 y 20 mm de longitud, estar en buen estado y libre de espinas. Además, deben estar libres de cualquier material extraño, especialmente en la abertura apical, y no deben mostrar signos de humedad externa anormal ni de olores o sabores extraños [20]. La Figura 4 se muestra una clase de pitahaya amarilla en buen estado.



Figura 4. Pitahaya amarilla para consumo [21].

En la Tabla 1 se muestra la clasificación según el calibre de la pitahaya. En el ámbito de la exportación, el calibre del producto se determina por la cantidad de frutos por unidad de empaque, mientras que, en el mercado nacional, se utiliza el calibre del producto para describir el rango de masas [20].

Tabla 1. Calibres de los frutos de acuerdo con la masa unitaria [20].

Calibre	Peso Unitario (gr)
8	> 361
9	261 a 360
12	201 a 260
14	151 a 200
16	111 a 150
20	< 110

Debido al tamaño de las frutas que presenta el ecotipo “Palora” se ha determinado una tabla de calibres para la comercialización con fines de exportación (Tabla 2) [21].

Tabla 2. Calibres para exportación de pitahaya amarilla “Palora”. Nota: Toma de FRAPELIVE S.A. [21].

Rangos de peso de la fruta(g)	Calibres
>500	5
401 – 500	6
351 – 400	7
301 – 350	8
261 – 300	9

c. Características de la pitahaya

Según la norma NTE INEN 2003:2005 establece los requisitos que debe cumplir la fruta destinada para consumo nacional y de exportación.

Los requisitos físicos de la pitahaya amarilla son [20]:

- La pitahaya debe estar entera y sin heridas
- Forma ovoidal
- Aspecto fresco y consistencia firme
- El tallo debe medir 15 mm a 20 mm de longitud
- Deben estar sanos libres de ataques de insectos y enfermedades que determinen la calidad interna de la pitahaya
- Deben estar libres de humedad externa producida por la etapa de postcosecha como es: selección, acopio, clasificación, empaque, almacenamiento y transporte.
- No debe tener sabor y olor extraño.

La pitahaya amarilla se clasifica en tres categorías como indica la Tabla 3:

Tabla 3. Categoría de la Pitahaya Amarilla [20].

Categoría	Descripción
Categoría extra	La pitahaya que se encuentra en esta categoría debe cumplir las características antes mencionadas, donde permiten un grado mínimo de

Categoría	Descripción
	alteraciones en la cáscara, siempre y cuando no afecte la apariencia de la pitahaya.
Categoría I	En esta categoría al igual de la anterior debe cumplir las características principales de la fruta, pero a su vez se admiten varios defectos que presente la fruta como son: <ul style="list-style-type: none"> • Deformación del fruto • Rozaduras cicatrizadas sin excederse 1 cm^2 con respecto al área total del fruto
Categoría II	Esta última categoría se encuentra las frutas que no puede clasificarse en las otras categorías, pero cumplen con las características de la pitahaya.

1.3.3 Selección de la cosecha

El cambio de color de verde a amarillo se convierte en el indicador más frecuentemente observado para determinar la madurez de la fruta. La recolección se realiza manualmente, utilizando gavetas o cestas con una capacidad máxima de 5 kg. Debido a la maduración desigual y la presencia de espinas en la planta, la recolección se convierte en la tarea más delicada dentro del proceso de cultivo [22].

Debido al tallo extremadamente corto, extraer la fruta de la vaina puede resultar difícil. Se debe evitar retorcer la fruta, ya que esto podría rasgar la piel. En su lugar, se recomienda utilizar tijeras o un cuchillo afilado para realizar una pequeña incisión en la vaina. Posteriormente, colocar la fruta cortada en una caja o cesta en lugar de dejarla en el suelo [22].

Para preservar la calidad de la fruta destinada a la exportación, es fundamental comprender el desarrollo floral y la fructificación, evitando la exposición al sol, ya que esto eleva la temperatura y acelera el proceso de maduración. En el momento de la cosecha, se aconseja seleccionar frutas que se encuentren en el mismo grado de maduración [22].

Cuando se piensa en la exportación, es esencial cosechar los frutos en la fase "pintona", aproximadamente un 25% maduros. Permitir que los frutos maduren en la planta los haría más susceptibles a plagas y enfermedades, además de disminuir significativamente su vida útil una vez que llegan al punto de venta [22].

a. *Madurez óptima de la pitahaya*

La pitahaya es susceptible al etileno, sin embargo, esta sensibilidad puede neutralizarse mediante el empleo de sales, como el permanganato potásico (KMnO₄). Es esencial evitar el contacto directo entre dicha sustancia y el producto. En la Figura 5 se indican los grados de maduración de la pitahaya amarilla que permiten su valoración para la exportación y el consumo nacional [21].

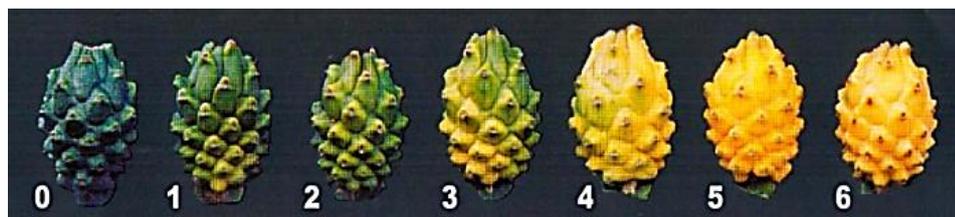


Figura 5. Grados de maduración de la Pitahaya Amarilla [21].

La maduración de la pitahaya se determina visualmente a través de su color externo, como se indica en la Tabla 4.

Tabla 4. Color de maduración de la pitahaya amarilla [21].

Color	Descripción
Color 0 (verde)	El fruto es de color verde cuando representa entre un 5% y 20% de su cáscara, pudiendo tener matices amarillos.
Color 1 (pintón)	Fruto que presenta un tono verde-amarillo, abarcando aproximadamente del 21% al 40% de su cáscara.
Color 2 (maduro)	Fruto que alcanza su totalidad de color amarillo, abarcando aproximadamente del 41% al 80% de su cáscara.

b. Técnicas de selección de la pitahaya

La cosecha de la fruta del dragón debe llevarse a cabo cuando el fruto alcanza un nivel de madurez comprendido entre $\frac{1}{4}$ y $\frac{1}{2}$, conocido como “pintona”, exhibiendo una apariencia y textura consistentes. Dada la maduración irregular de los frutos y la presencia de espinas en la planta, la recolección se realiza de forma manual, utilizando guantes y tijeras de podar curvas. Este método permite cortar los frutos justo en el pedúnculo sin causar daños a la rama. Posteriormente, los frutos se disponen en una lona o cesta y se transportan al centro de recolección, almacén o área de procesamiento. [21].

Tras haber completado la fundamentación de la producción de la pitahaya mediante la aplicación de diversos recursos y herramientas técnicas para mejorar su calidad, se

procederá a fundamentar teóricamente el uso de la visión artificial en el control de calidad de la pitahaya.

1.3.4 Inteligencia artificial (IA)

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para utilizar algoritmos, recopilar información a partir de datos y tomar decisiones de manera similar a un ser humano [23]. Esta disciplina de la informática se centra en dos aspectos fundamentales del comportamiento y razonamiento humanos: el razonamiento y los modelos computacionales basados en el comportamiento. Dichos modelos tienen la capacidad de realizar tareas que serían inalcanzables para los seres humanos [24].

a. Aprendizaje automático (Machine Learning)

El aprendizaje automático se posiciona como uno de los enfoques clave en el desarrollo de la inteligencia artificial. Fundamentalmente, esta disciplina informática se ocupa de investigar cómo las máquinas o computadoras pueden aprender sin depender de una programación específica. En contextos concretos, las recomendaciones o predicciones a menudo constituyen ejemplos típicos de los resultados logrados [23].

b. Redes neuronales profundas

Una entidad computacional denominada red neuronal se compone de un conjunto de elementos esenciales conocidos como neuronas artificiales, que se organizan típicamente en capas. Estas neuronas se conectan entre sí para facilitar el intercambio de señales, representadas mediante valores numéricos, como se ilustra en la Figura 6 [25].

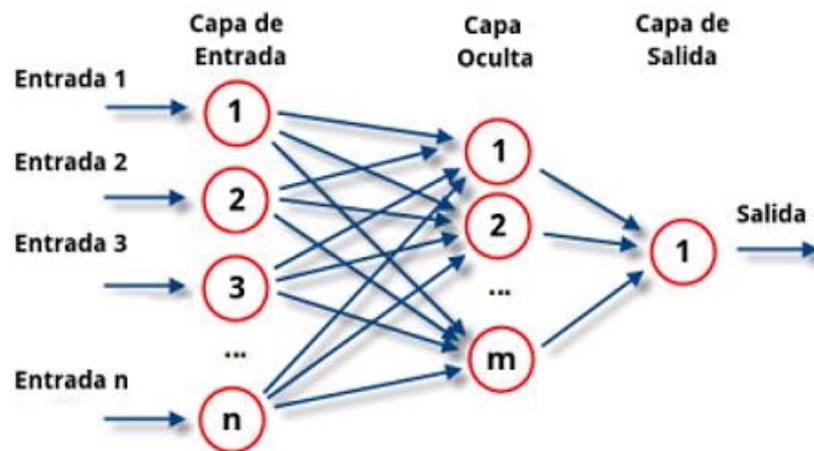


Figura 6. Ejemplo de red neuronal [25].

A través de estas conexiones, una neurona tiene la capacidad de modular la activación de las neuronas vecinas, ya sea suprimiéndola o potenciando la salida de la neurona anterior mediante un valor de peso. Las redes neuronales están diseñadas para abordar problemas de manera más abstracta que el cerebro humano [25].

1.3.5 Procesamiento de imágenes

En muchas ocasiones, el control de calidad utiliza instrumentos; sin embargo, todavía en la actualidad, muchos procesos de inspección dependen enteramente de la experiencia de un inspector y su habilidad visual para identificar defectos en los productos. Por lo general, estos análisis se llevan a cabo mediante métodos, ya sean digitales u ópticos, implementados en un sistema informático [26].

a. Segmentación semántica

Para aislar las áreas de interés del resto de la imagen, la segmentación divide la imagen en sus componentes individuales u objetos. El nivel de subdivisión puede variar según la naturaleza del problema. Durante el proceso de identificación de las partes de una imagen, se pueden reconocer los bordes o dividir la imagen en regiones, líneas, curvas, entre otros, como se muestra en la Figura 7 [27].



Figura 7. Ejemplo de binarización de una imagen [28].

b. Detección de objetos

El objetivo consiste en procesar una imagen que contenga varios objetos para que una computadora pueda identificar los distintos tipos de objetos presentes. Para lograrlo, se parte de una imagen con múltiples objetos y se mejoran sus características mediante la aplicación de filtros de suavizado. Posteriormente, se asigna un valor a cada objeto utilizando técnicas de detección de objetos [28]. El proceso por el cual un sistema automatizado utiliza una cámara para reconocer objetos específicos se conoce como detección de objetos mediante una cámara, como se ilustra en la Figura 8.

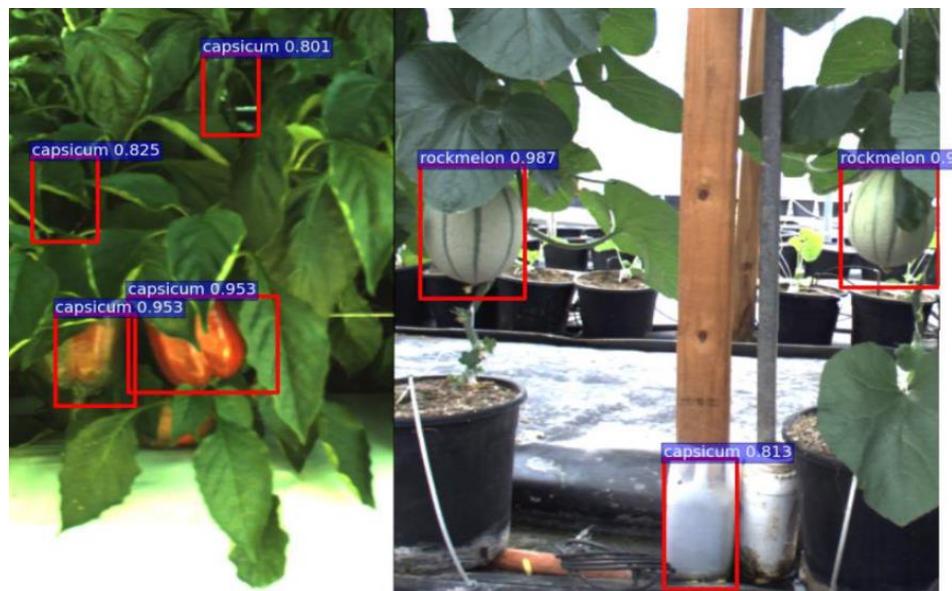


Figura 8. Resultados de detección de varias frutas [15].

1.3.6 Visión artificial

Los avances en la tecnología de visión artificial han capacitado a los equipos industriales para percibir y responder rápidamente a su entorno. Las aplicaciones más

comunes de la visión artificial incluyen la identificación, clasificación y seguimiento de productos, así como la colocación y medición de piezas. Además, se utiliza para la inspección visual y la detección de defectos [29]. Gracias a la visión artificial, los equipos industriales pueden observar y analizar tareas relacionadas con la fabricación inteligente, el control de calidad y la seguridad de los trabajadores, aprovechando las últimas tecnologías de inteligencia artificial. En la Figura 9 se presenta la arquitectura general de la Visión Artificial [30].

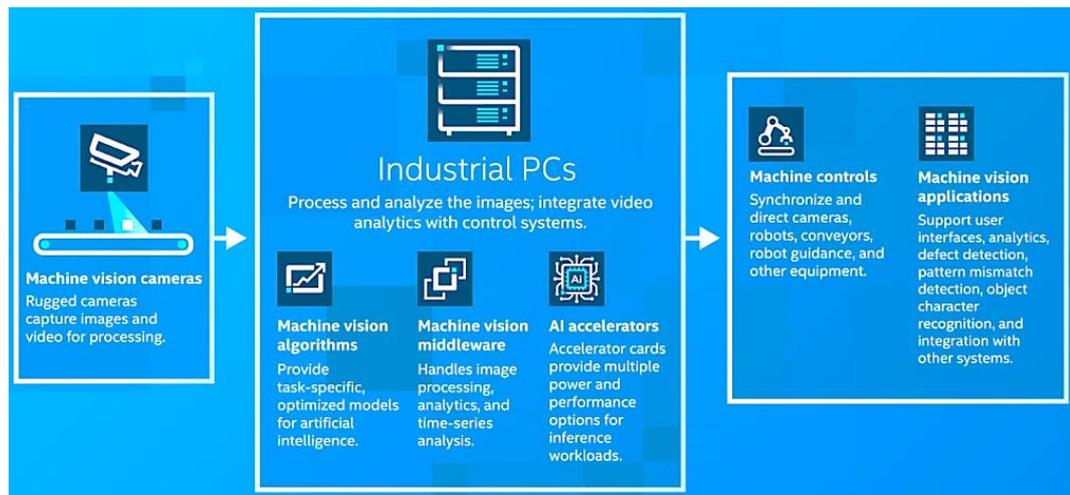


Figura 9. Visión Artificial [29].

a. Etapas de un proceso de visión artificial

La visión artificial comprende una amplia variedad de conceptos que abarcan tanto el hardware como el software, tal como se indica en la Tabla 5 y se representa en la Figura 10.

Tabla 5. Etapas de un proceso de Visión Artificial

Pasos	Descripción
1	El primer paso es el proceso es adquirir la imagen digital.
2	Una vez que se ha obtenido la imagen digitalizada, el siguiente paso implica el preprocesamiento de esa imagen
3	El paso siguiente es la segmentación. Definida en sentido amplio, su objetivo es separar la imagen en sus componentes individuales o en los objetos que la conforman
4	El resultado del proceso de segmentación es una imagen de datos que representa el límite de la región o sus puntos constituyentes. La elección inicial consiste en determinar si se utiliza la representación por límites o de toda la región.
5	Seleccionar una representación adecuada es solo uno de los pasos en el proceso de transformación de los datos de entrada. Además, es necesario definir un método que permita extraer los datos relevantes. La parametrización, también conocida como selección de características, se encarga de identificar características que proporcionen

Pasos	Descripción
	información cuantitativa de interés o que sean fundamentales para distinguir una clase de objetos de otra.
6	El reconocimiento y la interpretación vienen en último lugar. El reconocimiento consiste en etiquetar un objeto basándose en la información proporcionada por los descriptores (clasificación). A continuación, la interpretación asigna un significado al conjunto de objetos reconocidos.

En el ámbito de la visión artificial, la comprensión de las etapas de un sistema es esencial para apreciar la complejidad y funcionalidad de esta tecnología. El siguiente diagrama de bloques ofrece una representación visual de las diversas fases que componen un sistema de visión artificial. Desde la adquisición de la imagen hasta la interpretación de datos, cada bloque desentraña un componente clave en el proceso.

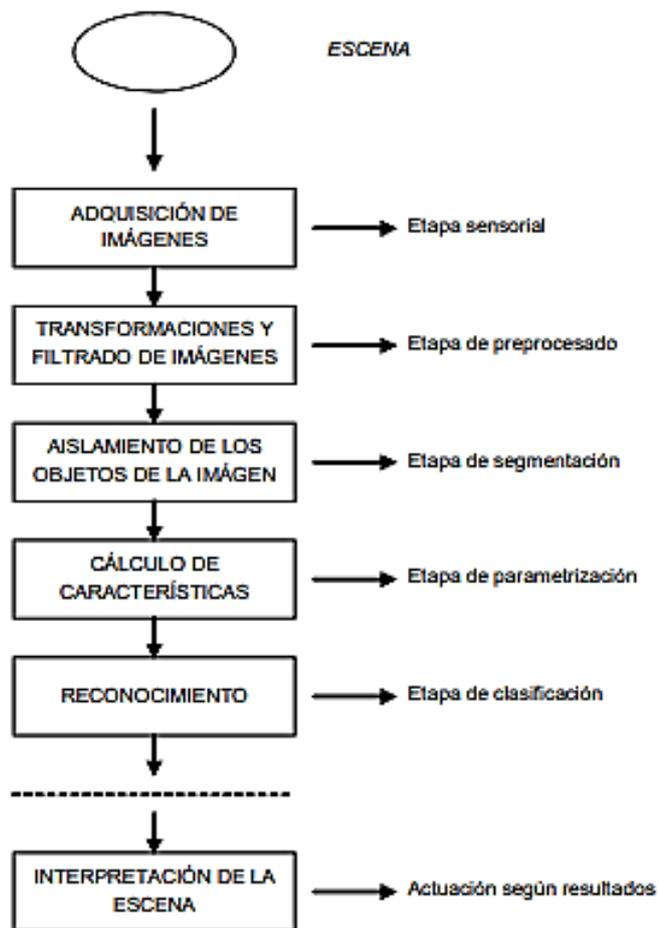


Figura 10. Diagrama de bloques de las etapas de un sistema de visión artificial [30].

b. Componentes de la visión artificial

Los elementos de hardware mínimos necesarios para un sistema de visión artificial (SVA) son los siguientes:

Sensor óptico. Se podría emplear una cámara en color o monocroma como sensor, capturando una imagen completa del área del sujeto cada 1/30 de segundo. Otra opción para los sensores sería una cámara de barrido que genere continuamente una línea. En este caso, la imagen bidimensional se forma al mover el objeto a lo largo de la línea del escáner, o viceversa. Las características específicas del sensor y la imagen resultante son determinadas por la aplicación en cuestión [31].

Tarjeta de adquisición de imagen. Simplifica la conversión de la señal de vídeo proporcionada por el subsistema anterior a formato digital [31].

Computador. Una vez transformada la imagen en formato digital, es crucial almacenarla en la memoria de un ordenador para su posterior procesamiento y manipulación mediante un programa informático [31].

Monitor de vídeo. Facilita la visualización tanto de las imágenes o escenas capturadas como de los resultados del procesamiento de dichas imágenes [31].

En la Figura 11 se presenta el objeto observado, la iluminación, la cámara y el procedimiento de escaneo. Este procedimiento facilita la visualización de la imagen digital y, simultáneamente, la extracción de parámetros que se someten a un análisis y control posteriores. Este análisis y control pueden influir en el entorno del objeto, ya sea para clasificarlo, descartarlo, tomar medidas u realizar otras acciones. [31].

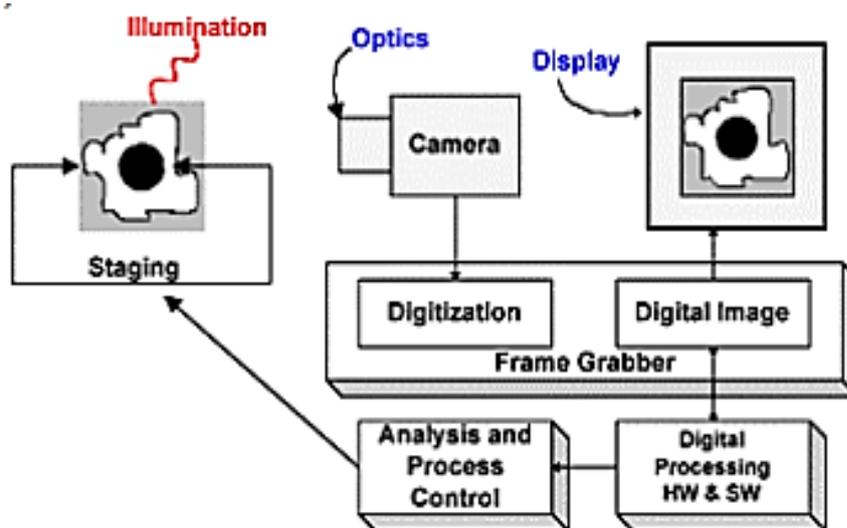


Figura 11. Diagrama de bloques de un sistema SVA [31].

c. Red Convolucional YOLO

YOLO, que significa 'You Only Look Once', es una red convolucional innovadora que realiza predicciones simultáneas de múltiples cuadros delimitadores y sus probabilidades de clase asociadas. A diferencia de la segmentación de imágenes, YOLO utiliza características de la imagen completa para hacer predicciones para cada cuadro delimitador. Esta red también tiene la capacidad de predecir cuadros delimitadores para todas las clases presentes en una imagen al mismo tiempo. YOLO divide la imagen de entrada en una cuadrícula $S \times S$ y asigna a cada celda de la cuadrícula la responsabilidad de detectar un objeto si su centro cae dentro de ella. Además, YOLO se entrena con imágenes completas, optimizando directamente el rendimiento de la detección. Una característica destacada es su capacidad para procesar vídeo en tiempo real con una latencia de menos de 25 milisegundos. En la Figura 12, se encuentra la arquitectura de YOLO. [32]

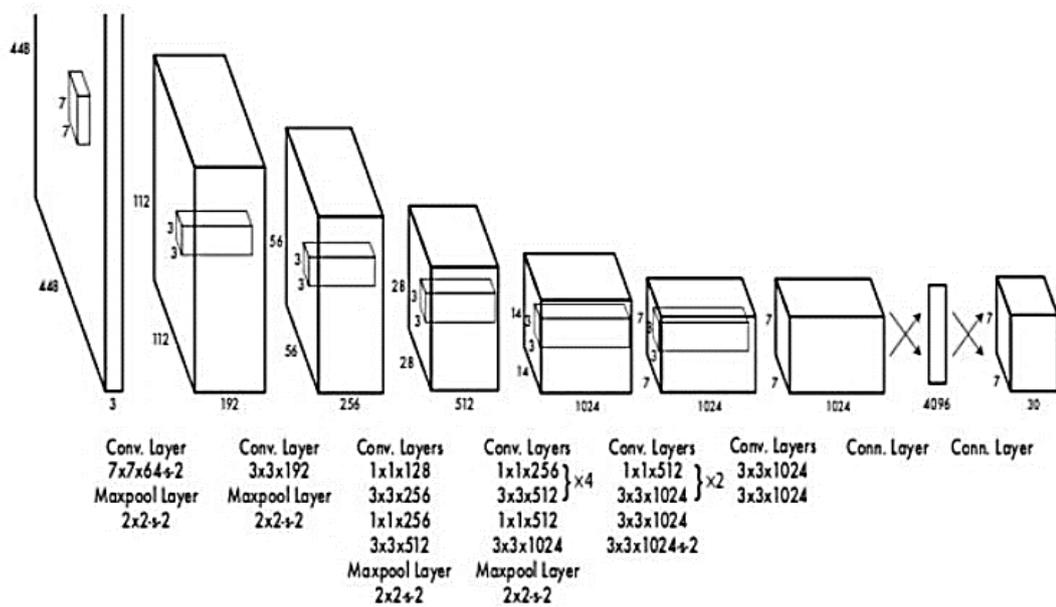


Figura 12. Arquitectura de la red convolucional YOLO [32].

d. YOLOv5

Este sistema, desarrollado como una solución de código abierto, se ha creado con el propósito de llevar a cabo la detección de objetos en tiempo real. Está equipado con un modelo preentrenado utilizando el conjunto de datos COCO y emplea una única red neuronal convolucional (CNN) para reconocer y clasificar objetos en las imágenes

[33]. En la Figura 13, se presenta la arquitectura general del modelo YOLOv5, que consta de una capa de entrada que se envía a una red troncal mediante tres mapas, llega a la cabeza donde se ejecuta el cálculo de confianza y regresión para así culminar con los umbrales del proceso NMS [34].

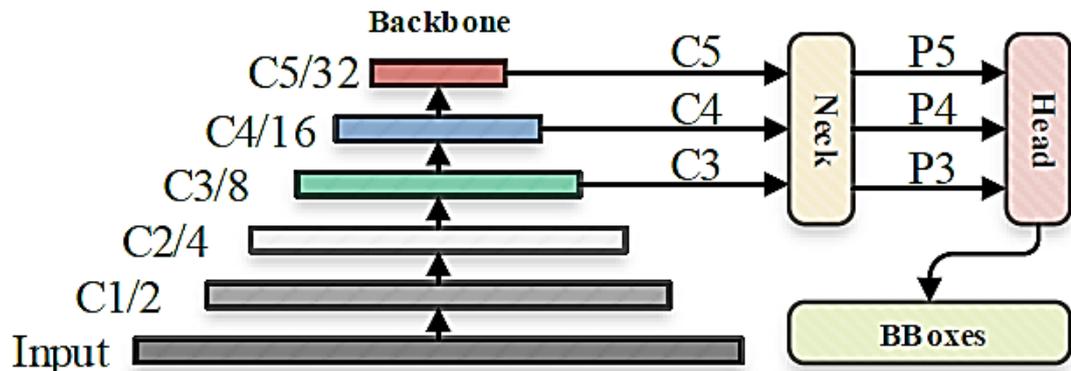


Figura 13. Arquitectura YOLOv5 [34].

e. *Makeense.ai*

Makesense.ai es una herramienta en línea diseñada para la etiquetación de imágenes. Se ofrece de manera gratuita y no necesita ser instalada, ya que opera directamente en el navegador. Es compatible con diversos sistemas operativos. Esta aplicación resulta especialmente destacada en proyectos de visión artificial, ya que agiliza y simplifica el proceso de preparación de conjuntos de datos [35].

f. *Base de Datos*

Una base de datos constituye una agrupación de información en un contexto particular, posibilitando el almacenamiento organizado de diversas cantidades de datos para su recuperación y uso eficiente. Diversas aplicaciones informáticas pueden emplearse para la creación y administración de bases de datos, algunas de las cuales se presentan en la Tabla 6 [35]:

Tabla 6. Base de Datos [35].

Software	Descripción
MYSQL	MySQL se destaca como uno de los sistemas de gestión de bases de datos más populares para aplicaciones web. Mantiene su carácter gratuito y experimenta una evolución constante con mejoras en sus características y seguridad. Este sistema demuestra eficientemente su capacidad para manejar grandes cantidades de datos sin generar una carga excesiva en los recursos del servidor, convirtiéndose en la

Software	Descripción
	elección ideal para desarrolladores independientes que buscan una solución robusta y ágil.
phpMyAdmin	PhpMyAdmin sirve como una herramienta para administrar servidores de bases de datos MySQL, ofreciendo una interfaz gráfica fácil de manejar y disponible de manera gratuita. A través de esta plataforma, los usuarios pueden realizar una variedad de operaciones en la base de datos, como crear, eliminar y modificar tablas, así como insertar, modificar y eliminar datos. Además, facilita la definición de usuarios, la asignación de permisos y la creación de copias de seguridad, entre otras funciones

g. Python

Este lenguaje de programación, siendo de código abierto, presenta la capacidad de trabajar eficientemente en diversas plataformas y se caracteriza por una semántica dinámica de alto nivel, lo que contribuye al desarrollo óptimo de aplicaciones. Su compatibilidad abarca tanto sistemas operativos Windows como Linux, aunque se ve limitada por sus versiones anteriores y su funcionalidad. La interfaz de programación de este lenguaje es dinámica, eliminando la necesidad de declarar el tipo de datos al inicializar variables. Destaca por su amigabilidad con los desarrolladores y permite optimizar el tiempo al programar de manera precisa. [36]

h. Análisis comparativo de en técnicas utilizadas en cultivos de frutas mediante visión artificial

En la Tabla 7 presenta un análisis comparativo de diversos métodos, algoritmos y modelos aplicados en cultivos de distintas frutas. Cada autor o grupo de autores propone sistemas que integran hardware y software para el control de calidad de las frutas, permitiendo que la visión por computadora opere de manera eficaz. Además, se detalla el número de muestras de imágenes utilizado en el entrenamiento y la eficiencia de cada uno de los sistemas respectivos.

Tabla 7. Técnicas utilizadas en cultivos de frutas por visión artificial [37] [10] [38] [39] [40] [41] [42] [15].

Autor	Título	Método /Algoritmo/Modelo	Equipos	# Muestras Entrenamiento	Eficiencia
Ledda Larcher Pedro Juárez Ana Ruggeri	Ponderación de calidad en frutas usando técnicas de visión artificial	Matlab e Image Processing Tolbox Algoritmo Recursive	Cámara digital	176 imágenes	74,34%

Autor	Título	Método /Algoritmo/Modelo	Equipos	# Muestras Entrenamiento	Eficiencia
Enrique Biasomi Carlos Catteneo Gustavo Villalba					
J. Herrera S. Medina K. Beleño O. Guadrón	Sistema automático de selección de frutos de café mediante técnicas de visión artificial	Algoritmos de procesamiento de imagen Python 2.7.8 Open CV 2.4.9 Método Otsu	Cámaras Iluminación LED blanca	-	87%
Diego Figueroa Edgar Roa	Sistema de visión artificial para la identificación del estado de madurez de frutas (granadilla).	Método Otsu OpenCv en Python	Módulo Picamera de Raspberry pi Iluminación direccional iluminación Led (diodos emisores de luz) con fondo negro opaco.	90 imágenes	92.6%
Luis Molina Carlos Vargas	Implementación de un sistema de control de calidad para la detección de contaminantes en diferentes tipos de frutas usando visión artificial	Python OpenCv TensorFlow	Cámara Web Iluminación artificial	845 imágenes	90%
Pedro Atencio	Sistema de visión artificial para el control de calidad del mango de azúcar de exportación	Modelo geométrico de volumen Método de error medio cuadrado	Cámara fotográfica Luz blanca	142 imágenes	99,29%
Brian Nieto José Rangel	Sistema de visión artificial para gestión de calidad del Banano Cavendish en etapa de postcosecha	Modelo de machine learning Python Algoritmos en C++	Raspberry Pi Camera Module V2 al Raspberry Pi	1093 imágenes	97,41%
Enoc Tapia	Sistema de alerta del estado de maduración de alimentos frescos dentro de un refrigerador utilizando	Modelos de aprendizaje profundo Python Jupyter Notebook Google Colab Librerías: Tensorflow	CPU Apple M1 con 8 núcleos y 16 GB de RAM Raspberry Pi 4 Raspberry Pi Camera	3900 imágenes	97,86%

Autor	Título	Método /Algoritmo/Modelo	Equipos	# Muestras Entrenamiento	Eficiencia
	Inteligencia Artificial	Pytorch Sklearn Pandas Numpy Matplotlib Modelo YOLOv7y v8			
Jeison Yarce	Estudio para la implementación de cosecha selectiva de café arábico aplicando vibraciones de alta frecuencia	OpenCV YOLOv3 YOLOv5 Mobilenet v2	-	-	94%
Andrés Garcés	Clasificación de calidad de manzana para monitoreo de cosechabilidad utilizando visión por computador y algoritmos de aprendizaje profundo	Modelo de red de detección SSD-MobileNet Red neuronal convolucional rápida FCN-ResNet 18	Jetson Nano Cámara Canon PowerShot SX400 IS	5125 imágenes	92,6%

1.4 Objetivos

1.4.1 Objetivo general

- Desarrollar un sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial.

1.4.2 Objetivos específicos

- Analizar los elementos de producción y características que determinan la calidad de la fruta.
- Determinar las técnicas actuales de Visión Artificial aplicado en cultivos.
- Aplicar un algoritmo de inteligencia artificial para el análisis de la calidad de la cosecha.

- Implementar una interfaz gráfica para la visualización de resultados del sistema inteligente de selección de cosecha.

CAPÍTULO II.- METODOLOGÍA

2.1 Materiales

El presente proyecto, se ha desarrollado un sistema inteligente para la selección de cosecha en el cultivo de la pitahaya. Este sistema tiene la capacidad de identificar y clasificar automáticamente las frutas aptas tanto para el consumo humano como para la exportación. Para llevar a cabo este proceso, se utiliza una cámara web TRUST GXT-1160 para la captura de imágenes, la cual está acompañada de iluminación led blanca artificial. La computadora ejecuta algoritmos que, posteriormente, almacenan los resultados en una base de datos. Todo este procedimiento se lleva a cabo dentro de una cabina que integra el sistema de visión artificial encargado del control de calidad de la pitahaya.

El uso de la plataforma gratuita Makeense.ai permite el etiquetado de las imágenes adquiridas para el entrenamiento mediante el modelo YOLOv5. La validación se lleva a cabo en la plataforma Google Colaboratory, la cual proporciona potencia para manejar grandes conjuntos de datos y realizar diversos entrenamientos. Además, se emplea el lenguaje de programación Python, que desempeñó un papel fundamental en la creación de la interfaz para el sistema de visión artificial.

2.2 Métodos

2.2.1 Modalidad de la investigación

Este proyecto se apoyó en la investigación aplicada ya que se aprovechó los conocimientos adquiridos durante la carrera y permitió realizar un análisis de la pitahaya en el proceso de maduración y cosecha de esta fruta mediante una solución a través de un sistema inteligente utilizando visión artificial.

Se realizó una investigación bibliográfica por que los contenidos científicos se buscaron a través de la recopilación de información de datos bibliográficos de libros, tesis anteriores, revistas, anuarios, publicaciones científicas, registros bibliográficos y estadísticas de informes.

Adicionalmente, se realizó investigaciones de campo para recopilar datos sobre las estrategias y técnicas empleadas para el control de calidad de la cosecha en cultivos de pitahaya que sean adecuadas para el mejor desarrollo del producto.

La investigación también fue de tipo experimental ya que se centra en la implementación de pruebas y experimentos controlados para evaluar el rendimiento y la efectividad de los sistemas de visión artificial en el análisis de la calidad de la pitahaya.

2.2.2 Recolección de información

Para la recolección de información se realizó la siguiente Tabla 8 donde se responde varias preguntas para tener un concepto de esta actividad.

Tabla 8. Recolección de información

Criterio	Descripción
¿Para qué?	Para clasificar la pitahaya en la época de cosecha.
¿De qué personas u objetos?	En los cultivos de pitahaya.
¿Como?	Algoritmos de visión artificial y procesamiento de imágenes
¿Con que?	Un sistema de cámaras mediante visión artificial
¿Cuándo?	En el periodo académico septiembre 2023-febrero 2024
¿Donde?	En el cantón Palora, vía a Sangay sector los Laureles

2.2.3 Procesamiento y análisis de datos

Para procesar y analizar la información se ha seguido los siguientes puntos:

- Búsqueda de información referente al tema
- Revisión de fuentes bibliográficas investigadas.
- Estudio de la metodología usada para la óptima elaboración de los sistemas.
- Planteamiento de la propuesta de solución.
- Control y verificación de los datos obtenidos mediante el funcionamiento del prototipo.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1 Análisis de la producción de pitahaya en el Ecuador

3.1.1 Producción de pitahaya en el Ecuador

Agrocalidad tiene 1891 productores registrados que están relacionados con la exportación del cultivo de pitahaya, sin embargo, sólo hay 7216 hectáreas de producción de esta fruta en Ecuador. Las provincias más productoras son Morona Santiago, Pichincha, Manabí y Guayas [43].

La pitahaya presenta dos temporadas de cosecha destacadas cada año: la primera tiene lugar entre los meses de febrero y marzo, y la segunda entre los meses de julio y agosto. Los meses de mayor cosecha en el Oriente ecuatoriano son febrero y marzo, que representan casi el 60% de la producción anual del país, mientras que los meses de junio, septiembre y noviembre, hasta los primeros días de diciembre, sólo se produce el 40% [19].

Morona Santiago controla el 65% de los lugares de producción registrados oficialmente en el país, seguida de Manabí (8%) y Pichincha (5,5%). De acuerdo con el Plan Operativo de Trabajo entre Ecuador y Estados Unidos, que también exige que toda la fruta exportada desde Ecuador sea procesada en instalaciones de almacenamiento registradas y certificadas como estructuras de exclusión de plagas, estos lugares de producción son monitoreados para prevenir la presencia de moscas de la fruta [44].

El mayor mercado de destino está en Estados Unidos, seguido de Singapur, Canadá y Colombia. Ya existen regulaciones fitosanitarias para exportar esta fruta a 56 países, lo que tiene un impacto positivo para los productores ecuatorianos al incrementar sus bases comerciales, oportunidades de empleo y exposición de sus productos a mercados globales. Mediante convenios entre el Ministerio de Agricultura y Ganadería de Ecuador (MAG) y el Servicio Nacional de Sanidad Agraria de Perú (Senasa), a través de la Agencia de Regulación y Control Fitosanitario y Zoosanitario (Agrocalidad), Ecuador exporta pitahaya a Perú desde noviembre de 2022 [45].

3.1.2 Ubicación

La finca productora de la pitahaya amarilla está ubicada en la provincia de Morona Santiago, cantón Palora vía a Sangay sector los Laureles. La ubicación exacta se encuentra en la Figura 14.

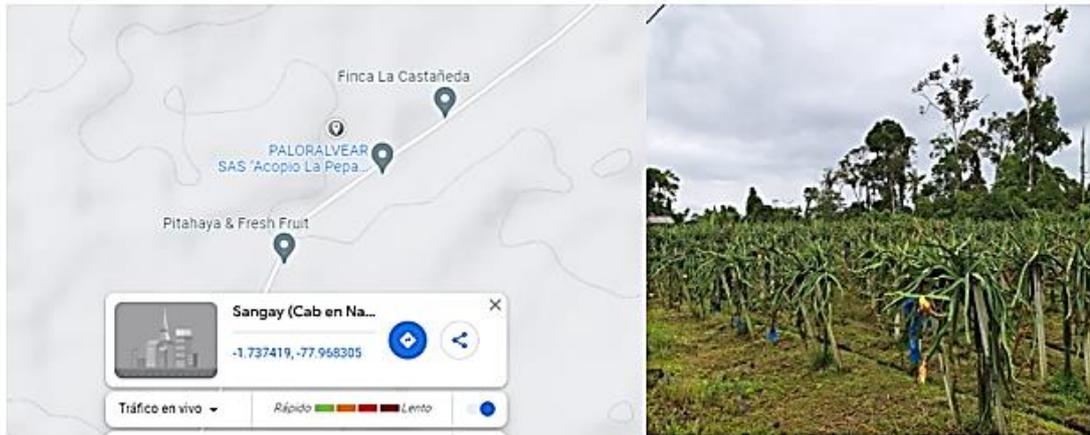


Figura 14. Mapa de ubicación de la finca (Google Maps).

3.1.3 Operaciones básicas de cosecha de la pitahaya

En la

Figura 15 se encuentra los pasos que se debe realizar para el proceso de cosecha de la pitahaya el cual se basó en [22]. La fruta debe pesarse cuando llega al centro de acopio para determinar la cantidad de producto disponible. Para eliminar la suciedad, las espinas, los restos de flores secas y las hormigas de los frutos, se lava con agua y cepillos suaves. A continuación, los productos se clasifican por tamaños y grados de madurez.

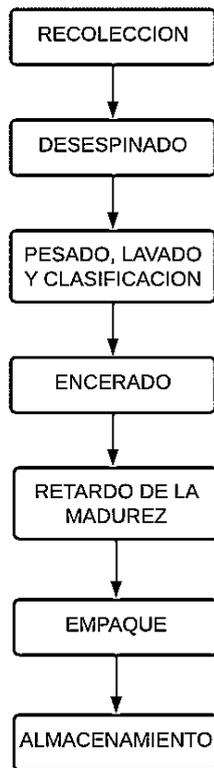


Figura 15. Proceso de cosecha de la pitahaya.

3.1.4 Exportación

Según la NORMA TECNICA ECUATORIANA NTE INEN 2003:2005 que se encuentra en el Anexo B, el proceso de recolección de la fruta se adapta a las demandas del mercado al que se destina. En el contexto de exportación, se prefiere la cosecha en el estado tres y cuatro de madurez, mientras que para consumo interno se opta por los estados cinco, como indica la Figura 16 [21].



Figura 16. Grados de maduración de pitahaya.

La calidad de los frutos está sujeta a rigurosas normas, que incluyen características como forma ovoidal, integridad, frescura y ausencia de daños por insectos o enfermedades. La pulcritud, la ausencia de espinas y de materias extrañas visibles son requisitos esenciales, junto con un pedúnculo de 15 a 20 mm de longitud y la ausencia de olores y sabores extraños [21].

En el Anexo A vinculado a los campos de cosecha y centro de acopio de pitahayas, se interpreta la clasificación para la exportación. Según el grado de maduración, los grados uno y dos están aptos para la exportación a Estados Unidos, el grado cuatro para la exportación a países vecinos (Perú, Colombia, Venezuela). Los grados cinco y seis se reservan para consumo nacional. La norma INEN 2003:2005 establece una clasificación adicional según calibres, que varían de ocho a veinte, determinados por la masa unitaria en gramos. Estos valores definen el peso por pitahaya para el mercado local y el número por cartón para la exportación.

Los calibres para la comercialización de la pitahaya con fin de exportación se presentan en la siguiente Tabla 9 [21].

Tabla 9. Calibres para exportación de pitahaya amarilla “Palora” [21].

Rangos de peso de la fruta(g)	Calibres
>500	5
401 – 500	6
351 – 400	7
301 – 350	8
261 – 300	9

Esta meticulosa regulación asegura que la pitahaya ecuatoriana cumpla con estándares de calidad internacionales, garantizando su aceptación en diversos mercados y su disfrute por consumidores tanto nacionales como internacionales [21].

3.2 Arquitectura del Sistema

En la Figura 17, se muestra la arquitectura general del sistema inteligente de selección de cosecha en cultivo de pitahaya constituido por tres etapas descritas a continuación.

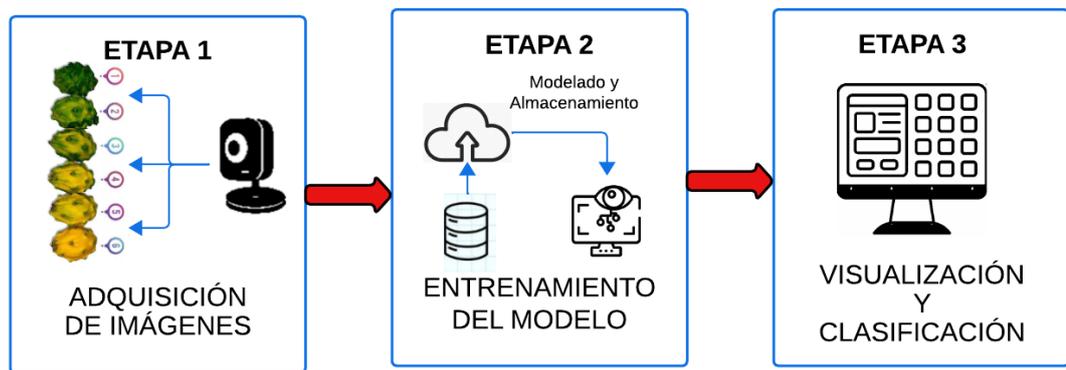


Figura 17. Esquema General Sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial.

Etapa 1.- Adquisición de imágenes

Esta etapa se enfoca en realizar la captura de fotografía de la pitahaya mediante una cámara web previamente calibrada, la cual capta el color y tamaño de la fruta para la siguiente etapa.

Etapa 2.- Entrenamiento del modelo

Se utiliza una base de datos en conjunto de algoritmos de visión artificial que permite entrenar al modelo con el control de calidad de la pitahaya para exportación y consumo nacional.

Etapa 3.- Visualización y Clasificación

La visualización mediante una interfaz gráfica permite el análisis de la pitahaya ya sea en el tamaño, y nivel de maduración para la respectiva clasificación estos resultados se muestran en una interfaz gráfica amigable con el usuario.

3.2.1 Selección de hardware

La selección de los elementos de hardware para el desarrollo del sistema se lleva a cabo teniendo en cuenta los parámetros y requerimientos del mismo, con el fin de establecer el elemento que mejor se adapte al prototipo. El objetivo del sistema requiere la adquisición de imágenes en tiempo real de la pitahaya por lo cual lo

primordial de los dispositivos deben tener alcance, resolución, tipo de imagen y envío de datos adecuado para el sistema de visión artificial [46].

Tabla 10. Selección de cámara web [47] [48] [49].

Características	Trust GXT1160	Web Cam D-bug M26	Pi Noir V2
Tamaño	20x10x50mm	19x7,6x12,2 cm	25x24x11,5mm
Resolución	3840x2160 pixeles		3280x2464 pixeles
Tasa de velocidad	30 fps	30 fps en modo USB	
Conexión	USB versión 2.0 Cable 150 cm de largo	USB versión 2.0	Conector de cámara
Iluminación	No	Desde 2 lux hasta 150 lux	-
Software compatible	Windows, macOS, Linux	Windows, macOS, Linux	Raspberry Pi, Linux
Ángulo de visión	55°	No especificado	62, 2°(horizontal)
Costo	\$25,00	\$4,50	\$35,00

Basándose en las características de las cámaras examinadas en la Tabla 10, se elige la Cámara Web Trust GXT-1160 con el datasheet del Anexo C, ya que puede conectarse a un ordenador con facilidad, ofrece un amplio ángulo de visión, es lo bastante ligera y pequeña como para caber en el prototipo, y es lo bastante versátil como para utilizarse en diversas aplicaciones. Una mayor resolución en píxeles indica que la calidad de la imagen es superior.

3.2.2 Sistema de Iluminación

La adquisición de imágenes debe tener lugar en un entorno determinado, por lo que la iluminación es crucial, ya que afecta a las expectativas de la imagen. Un adecuado sistema iluminación posibilita simplificar el procesamiento y análisis de imágenes mediante algoritmos de visión por computadora [50]. En la Tabla 11 se analiza las diferentes opciones de iluminación para el sistema de visión artificial.

Tabla 11. Sistema de iluminación [51] [52] [53].

Características	Led	Lámparas fluorescentes	Incandescentes
Eficiencia energética	Alta	Moderada a Alta	Baja a Moderada
Emisión de calos	Baja	Moderada	Alta
Calidad de luz	Muy alta	Moderada a Alta	Baja a Moderada
Tiempo que demora en encender	0,5 s	0,1 s	0,23 s
Vida útil	Hasta 50000 h	De 10000 a 20000 h	Alrededor de 1000 h

Características	Led	Lámparas fluorescentes	Incandescentes
Consumo energético	3,5, 5, 25 W	5, 7, 9 W	25, 50, 100 W
Emisión de color	2700K	3000K	2800K
Tono de luz	Blanca cálida y blanca fría	Blanco cálido, blanco, blanco frío, luz del día.	Luz muy cálida

Los tres tipos de iluminación artificial sugeridos se muestran en la Tabla 11 junto con la información técnica pertinente. La elección de la iluminación LED sobre la fluorescente e incandescente se basa en varias ventajas que las luces LED como indica el Anexo D ofrecen en términos de eficiencia, control, durabilidad y características específicas que son beneficiosas para aplicaciones de visión artificial.

3.2.3 Selección de software

a. Software de Desarrollo

El software de Desarrollo es un conjunto de herramientas, bibliotecas y entornos de programación que están diseñados para facilitar la creación, implementación y optimización de algoritmos y aplicaciones [54]. En la Tabla 12 se describen las características del software de desarrollo más conocidos Python, C++ y Java.

Tabla 12. Software de Desarrollo [36] [55] [56].

Características	Python	C++	Java
Tipo de lenguaje	Programación orientada a objetos, procedural y funcional	Programación orientada a procedimientos	Programación orientada a procedimientos
Nivel de lenguaje	Alto nivel	Alto nivel	Alto nivel
Sistema operativo	Windows, macOS y Linux	Windows, macOS, Linux	Windows, macOS, Linux
Desarrollo web	Utilizado en desarrollo web con frameworks como Django y Flask	Desarrollo web, especialmente con frameworks como Node.js y Express.js	desarrollo web, especialmente con frameworks como Spring
Plataforma	Independiente	Independiente	Independiente
Portabilidad	Altamente portátil, compatible con múltiples plataformas	Depende de la implementación y compilador	Altamente portátil a través de la Máquina Virtual Java (JVM)
Seguridad de datos	Seguridad menor que java	Muy Seguro	Muy seguro

Después de analizar las características de los tres softwares de desarrollo, se optó por seleccionar Python para el desarrollo de la aplicación debido a su compatibilidad con una amplia variedad de hardware y sensores. Python es reconocido como uno de los

lenguajes de programación más completos y de alto nivel, destacándose por su estructura de programación que gestiona la memoria automáticamente, lo que resulta en una menor cantidad de líneas de código. Esto facilita significativamente la integración con dispositivos específicos utilizados en el sistema de selección de cosecha.

b. Aplicación para código fuente

Proporciona un entorno integrado y unificado para escribir, editar, compilar, depurar y gestionar código, así como para llevar a cabo otras tareas relacionadas con el desarrollo de software.

Tabla 13. Aplicación para código fuente [57] [58].

Características	MyCompiler	Google colaboraty
Entorno de ejecución	En línea, basado en web	En la nube, en servidores de Google
Lenguajes de Programación	Soporte para varios lenguajes como C, C++, Java, Python, etc.	Principalmente centrado en Python, pero admite múltiples lenguajes.
Acceso a Bibliotecas	Limitado a las bibliotecas disponibles en el entorno en línea.	Acceso a un amplio conjunto de bibliotecas de Python, y la posibilidad de instalar otras adicionales.
Interfaz de usuario	Interfaz de usuario basado en la web	Interfaz de usuario basada en el navegador, similar a Jupyter Notebooks
Colaboración en tiempo real	No es nativamente compatible	Soporte para colaboración en tiempo real con otros usuarios.
Acceso a datos externos	Depende de la configuración local y disponibilidad de datos	Fácil acceso a conjunto de datos públicos por medio de Google Drive y servicios en la nube
Soporte para-GPU/TPU	Limitado por la capacidad del servidor en línea.	Soporte integrado para-GPU y TPU en la nube para tareas intensivas de procesamiento.
Costo	Ofrece planes gratuitos y premium.	Google Colab es gratuito con acceso a recursos de GPU y TPU, pero con limitaciones de tiempo de ejecución.

Una vez analizadas las características en la Tabla 13 de Google colaboraty y MyCompiler se seleccionó la primera debido a que proporciona recursos gratuitos de GPU y TPU, esenciales para tareas intensivas de procesamiento de imágenes en proyectos de visión artificial. Además, Google Colab cuenta con bibliotecas y frameworks preinstalados, permitiendo el acceso rápido a conjuntos de datos en la

nube, como los proporcionados por Google (por ejemplo, TensorFlow Datasets, Yolo), sin necesidad de descargarlos localmente. La integración con Google Drive simplifica el almacenamiento y acceso a datos, modelos entrenados y otros recursos.

c. Sistema de entrenamiento del modelo de visión artificial

Sistema de entrenamiento son un conjunto de procesos y herramientas utilizados para enseñar a un algoritmo a reconocer y localizar objetos específicos dentro de imágenes o vídeos [32].

Tabla 14. Sistema de entrenamiento del modelo de visión artificial [59] [60] [61].

Características	Yolov5	OpenCV	Tensorflow
Desarrollador/Contribuyente	Desarrollado por Ultralytics	Desarrollado por la comunidad de código abierto	Desarrollado y mantenido por Google
Enfoque	Detección de objetos en tiempo real	Librería de visión por computadora	Librería de aprendizaje automático completa
Arquitectura	Red Neuronal Convolutiva (CNN)	Varias funciones de procesamiento de imágenes	Plataforma para desarrollo y entrenamiento de modelos de machine learning, donde se incluye CNN.
Objetivo principal	Detección de objetos en imágenes y videos	Procesamiento de imágenes y visión por computadora	Plataforma para desarrollo y entrenamiento de modelos de machine learning en general.
Preentrenado	Sí, modelos preentrenados disponibles	No, OpenCV no se centra en modelos preentrenados	Sí, modelos preentrenados disponibles
Lenguaje de programación	Principalmente Python	Soporta varios lenguajes, incluyendo Python	Principalmente Python, pero también admite otros lenguajes a través de su API.
Facilidad de uso	Fácil de usar y desplegar	Fácil de usar para procesamiento de imágenes básico	Es utilizado de manera amplia, con documentación extensa.

Tras analizar las características en la Tabla 14 de los modelos de entrenamiento, se optó por seleccionar YOLOv5 para el desarrollo del sistema. Esto se debe a que YOLOv5 ha demostrado ser más estable y es altamente apreciado por su rendimiento y facilidad de uso en tareas de detección de objetos, garantizando eficiencia y precisión en la identificación de objetos en tiempo real. Este modelo ha probado ser

particularmente efectivo en tareas de visión artificial que involucran la identificación y localización simultánea de múltiples objetos, aspecto crucial para la selección precisa de cosecha en un cultivo de pitahaya. Además, la elección de este modelo se respalda en los resultados de la Tabla 7, donde se evidencian las mejoras en las capas ocultas de YOLOv5 y la incorporación de nuevas funciones que antes no existían.

d. Base de datos

Una base de datos es un sistema organizado para recopilar, almacenar y gestionar datos de manera estructurada, permitiendo el fácil acceso, gestión y actualización de la información.

Tabla 15. Base de datos [62] [63].

Características	MySQL	SQL Server	PostgreSQL
Desarrollador	Oracle Corporation	Microsoft Corporation	PostgreSQL Global Development Group
Licencia	GLP (Open Source)	Propietario (Comercial/Open Source)	PostgreSQL (Open Source)
Plataformas soportadas	Windows, Linux, macOS, BSD, Solaris, entre otros.	Windows, Linux	Windows, Linux, macOS, BSD, Solaris, entre otros.
Escalabilidad Horizontal	Sí	Sí	Sí
Seguridad	Funciones de autenticación y autorización estándar, control de acceso gratuito granular	Integración con sistemas de autenticación de Windows, control de acceso granular	Control de acceso granular, autenticación de contraseñas y autenticación de certificado.
Costo	Gratuito (Open Source), opciones comerciales disponible	Pago (licenciamiento por núcleo), versión Express gratuita disponible	Gratuito

Después de analizar las características de las tres bases de datos en la Tabla 15 se optó por MySQL debido a su interfaz gráfica de usuario que simplifica la gestión y manipulación de bases de datos, eliminando la necesidad de utilizar comandos manuales. Además, MySQL es un sistema de base de datos ampliamente popular y utilizado en todo el mundo, con una abundancia de información y tutoriales disponibles que lo hacen especialmente adecuado para proyectos de pequeña y mediana escala.

3.3 Diseño del Sistema

3.3.1 Entrenamiento del Sistema

Para realizar el entrenamiento las pitahayas fueron adquiridas de la finca productora de esta fruta se capturaron fotos de los grados de maduración de la pitahaya amarilla se emplearon 735 imágenes que se distribuyeron en conjuntos para el entrenamiento, validación y pruebas de acuerdo con las etapas de maduración de la fruta. En la Figura 18 se muestra almacenadas las imágenes realizadas, para la etapa 1 se emplearon 268 imágenes, para la etapa 2 se emplearon 152 imágenes y finalmente en la etapa 3 se utilizaron 315.



Figura 18. Imágenes de los grados de maduración de la pitahaya amarilla.

Para la adquisición de las fotografías se utilizó un dispositivo móvil: Infinix note 30 pro con cámara de 108 megapíxeles (gran angular) de resolución. En cuanto a la iluminación se utilizó una lampara led de 5500-6500K de temperatura del color como luz artificial constante durante todo el proceso de entrenamiento como se muestra en la Figura 19.

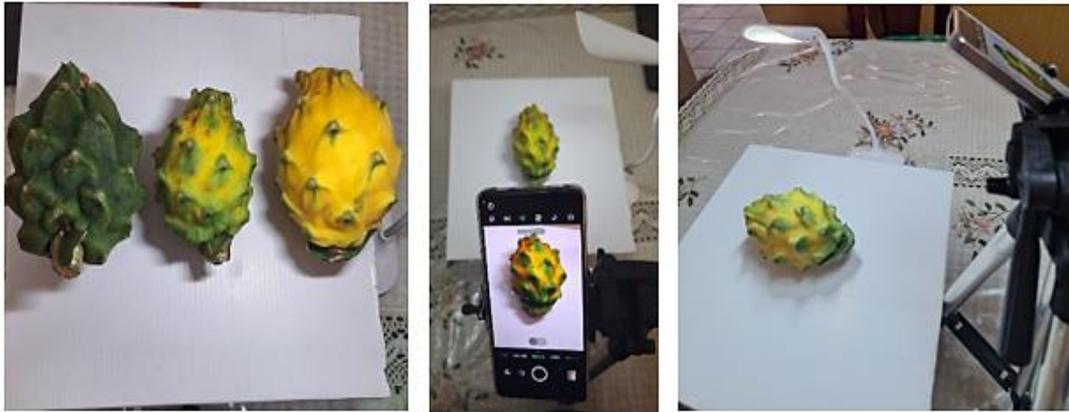


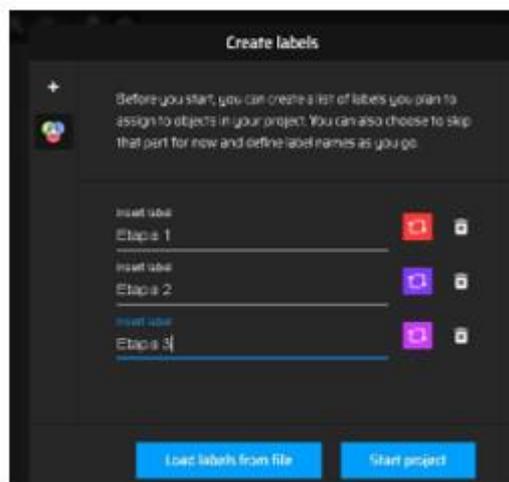
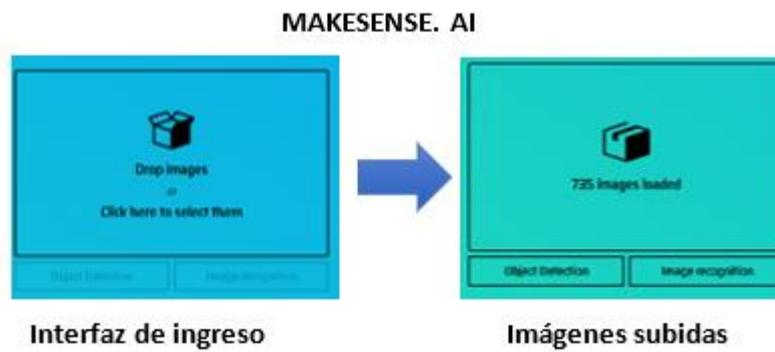
Figura 19. Proceso de adquisición de las imágenes para entrenamiento.

a. Etiquetado de las imágenes

Como primer paso se realiza el etiquetado de imágenes empleando el software makesense.ai que es una herramienta en línea de uso gratuito para etiquetar fotografía. Por medio de esta herramienta se puede realizar la anotación de objetos de interés de la imagen, lo cual implica la identificación y delimitación precisa de la pitahaya y sus características en las fotografías. El propósito fundamental de este etiquetado es la generación de un datasheet, es decir, un conjunto de datos con información detallada sobre las características de la pitahaya en cada imagen.

La generación de este conjunto de datos etiquetados es crucial para el desarrollo del sistema inteligente, debido a que permite al modelo de visión artificial aprender y comprender las características específicas de cada una de las etapas de maduración de las pitahayas. Este dataset se convierte en la base sobre la cual se construye el conocimiento del sistema, facilitando así su capacidad para realizar inferencia precisa durante las fases posteriores del proceso.

En la Figura 20, se muestra la interfaz de ingreso a la plataforma ya mencionada, se selecciona la opción Get Started para empezar con la creación del modelo. En esta ventana se debe seleccionar el total de las imágenes que se tiene que para el caso de la presente investigación son 735 imágenes. Para ingresar los nombres que se va a identificar en cada etapa de la pitahaya se debe seleccionar la opción object detection. En el desarrollo del proyecto se identificaron 3 etapas que son las que se visualizan en la siguiente imagen.



Nombre por etapas de las imágenes

Figura 20. Interfaz, selección de imágenes y etiquetado ya sea por grados o etapas

Una vez dentro del sistema se procede a seleccionar una por una todas las imágenes que pertenecen a cada etapa como se muestra en la Figura 21. Para la etapa 1 se emplearon 268 imágenes, etapa 2 se emplearon 152 imágenes de pitahayas, etapa 3 se utilizaron 315 imágenes de pitahaya maduras con un total de 735 imágenes etiquetadas.

PROCESO DE SELECCIÓN DE LAS IMÁGENES DE CADA GRADO DE MADURACIÓN

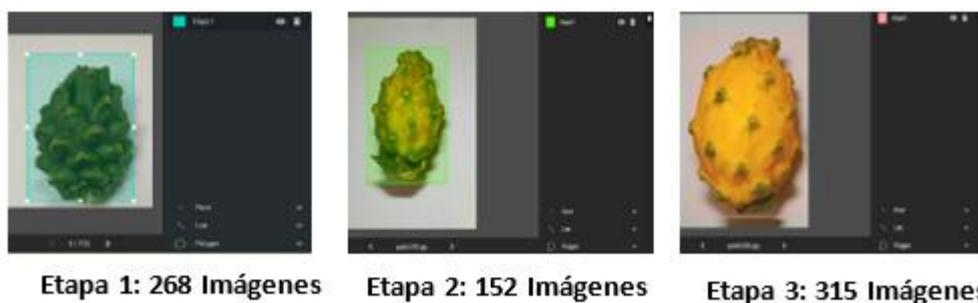


Figura 21. Selección de las imágenes de la pitahaya para cada grado de maduración

Una vez que se tiene las imágenes etiquetadas y seleccionadas se procede a importar los label generados con formato del modelo YOLO el cual nos da los labels como se muestra en la Figura 22.

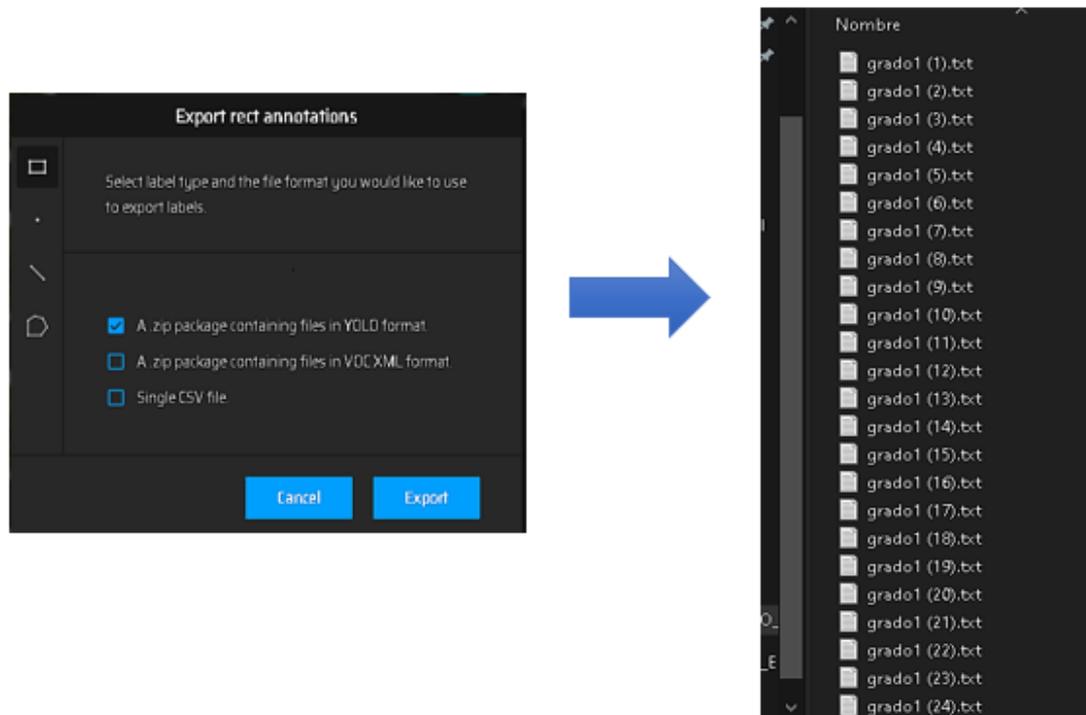


Figura 22. Labels generados.

b. División del conjunto de datos

En la Figura 23, se procede a separar los archivos para entrenar con 600 imágenes y de validación de 135 imágenes para ello se separa en carpetas a) carpeta principal con nombre de data, b) dentro se tiene dos subdirectorios una para images y la otra para labels a su vez c) se crea subdirectorio para entrenar y para validar (train y val). Una vez creadas las carpetas en la carpeta de images /train se coloca las 600 imágenes y en images/val las 135 imágenes para validar.

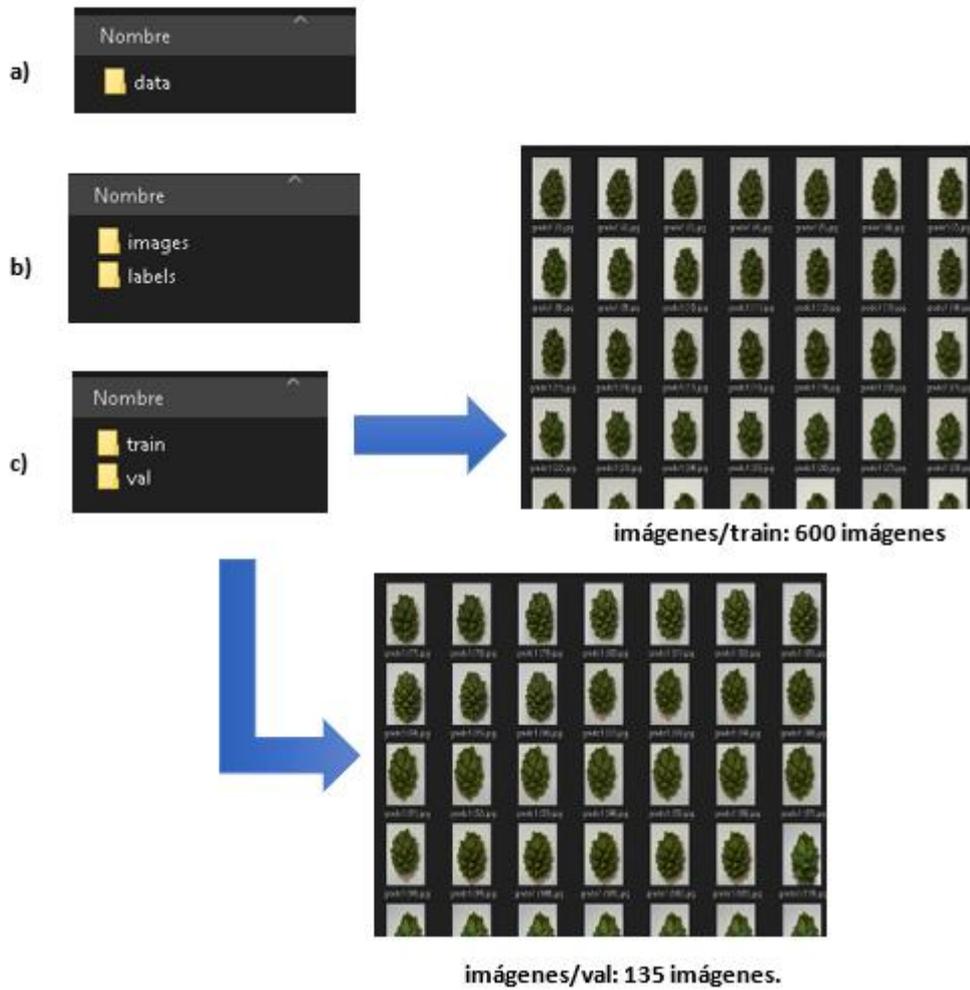


Figura 23. Creación de carpetas para entrenamiento y validación.

Dentro de labels/train se coloca los 600 labels y labels/val los 135 labels restantes como indica la Figura 24.

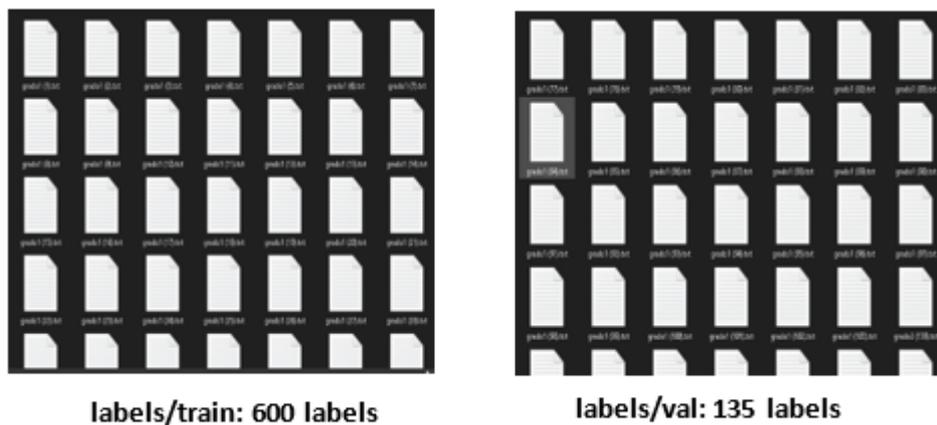


Figura 24. Labels de entrenamiento y validación.

Con el dataset debidamente creado, se procede a la generación de la estructura de carpetas específicas para el proyecto del sistema inteligente de selección de cosecha en el cultivo de pitahaya mediante visión artificial. Esta estructura comprende tanto las imágenes como las etiquetas distribuidas en subdirectorios correspondientes a los conjuntos de entrenamiento, prueba y validación.

Se procede a comprimir la carpeta principal (data) en formato zip, dicha carpeta se le sube a la nube Google drive para poder importar desde Google colaboraty. En la Figura 25 indica los cuatro pasos a realizar para seguir con la aplicación del algoritmo para el entrenamiento y validación del sistema de control de calidad de la pitahaya.

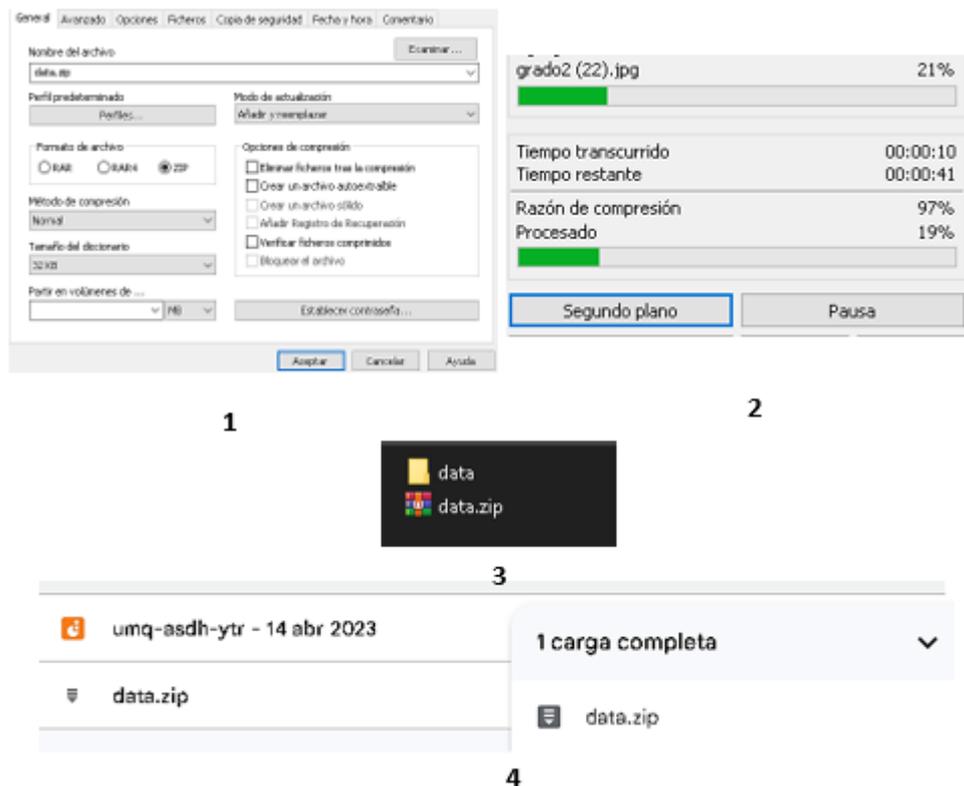


Figura 25. Compresión de la carpeta data y subida a Google Drive.

c. Código para el entrenamiento

Para el desarrollo del presente proyecto se empleó la plataforma en línea Google Colab, que ofrece acceso a una unidad de procesamiento gráfico (GPU) que se activa para agilizar los procesos de entrenamiento. Además, se utiliza Google Drive como repositorio principal para almacenar los códigos utilizados, modelos en desarrollo,

imágenes de entrenamiento, resultados obtenidos y demás archivos de importancia para el proyecto.

En primera instancia se sube el dataset en formato zip con las carpetas que se crearon al drive. Este paso asegura que el procedimiento sea eficaz y, de ser necesario ejecutar el cuaderno nuevamente. Al volver a abrir el cuaderno, es importante ejecutar las celdas para garantizar la correcta ejecución del conjunto de datos almacenados en Google Drive.

```
from google.colab import drive
drive.mount('/content/drive')
```

Para interactuar con Google Drive desde el entorno de Google Colaboraty, se utiliza el módulo “drive” que ofrece diversas funciones y métodos. Las rutas se pueden utilizar para acceder a los archivos de Google Drive una vez que se ha montado en el entorno Colaboraty.

Para descomprimir la carpeta almacenada con las imágenes y labels de la pitahaya se utiliza el comando `!unzip` descomprime el archivo ‘data.zip’ que se encuentra en Google Drive (`/content/drive/MyDrive/`) y coloca el contenido descomprimido en el directorio ‘`/content/`’ en Google Colab.

```
!unzip -q /content/drive/MyDrive/data.zip -d /content/
```

Se instala la librería `clearml` para poder acceder a esta plataforma externa que permite controlar el proceso de entrenamiento a su vez se guarda todo el entrenamiento y genera el modelo.

```
pip install clearml
```

En la página Códigos de acceso a la plataforma `Clearml` se genera un nuevo proyecto y se indica el acceso y llave para poder conectarse. La conexión se establece mediante el comando “`task.init`” con el nombre del proyecto creado.

```

%env CLEARML_WEB_HOST=https://app.clear.ml
%env CLEARML_API_HOST=https://api.clear.ml
%env CLEARML_FILES_HOST=https://files.clear.ml
%env CLEARML_API_ACCESS_KEY=Q2H2D838MPKP6F48AQ79
%env CLEARML_API_SECRET_KEY=KuGbJf7N1HS0DX3yibeEUkZceGD7geKWsngjM6kFoFnGYLFm1Q

from clearml import Task
task = Task.init(project_name="pitajaya", task_name="pitajaya")

```

Antes de iniciar el proceso de entrenamiento es necesario instalar los repositorios en Google Colab. Este paso implica la descarga de la carpeta principal del modelo YOLOv5 desde Github, con el propósito de instalar los requisitos necesarios proporcionados por estos archivos. Este procedimiento garantiza que todas las dependencias y configuraciones necesarias estén presentes en el entorno Colab, facilitando así la ejecución exitosa del entrenamiento del modelo Yolov5.

```

!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
%pip install -qr requirements.txt # install

import torch
import utils
display = utils.notebook_init() # checks

```

d. Generar entrenamiento

El comando de entrenamiento de YOLOv5, el cual establece parámetros esenciales, como el tamaño de las imágenes, el tamaño del lote, el número de épocas y la configuración del conjunto de datos y del modelo, para entrenar un modelo de detección de objetos.

```

!python train.py --img 400 --batch 35 --epochs 200 --data pitahaya.yaml --weights yolov5x.pt --cache

```

A continuación, se realiza una descripción de los comandos empleados en el entrenamiento del modelo:

- **img 400:** Especifica el tamaño de las imágenes que se utilizan durante el entrenamiento. En este caso, se establece en 400 píxeles que es la resolución de las imágenes de entrada.

- **batch 35:** Indica el tamaño del lote durante el entrenamiento. En el presente proyecto se establece en 35, lo que significa que se actualizarán los pesos del modelo después de procesar 35 imágenes.
- **epoch 200:** Establece el número de épocas de entrenamiento. Cada época implica pasar por todo el conjunto de datos una vez. En el presente proyecto se configuró para 200 épocas.
- **data:** Especifica el archivo YAML que contiene la configuración del conjunto de datos y del modelo. En el desarrollo del proyecto se llama pitahaya.yaml, este archivo tiene las etiquetas que se van a entrenar, por ejemplo, etapa1, etapa2 y etapa3.
- **weights:** Indica que se utilizarán pesos preentrenados para ahorrar tiempo de entrenamiento, yolov5x.pt en este caso.
- **Cache:** El cache de imágenes se realiza para llevar a cabo un entrenamiento rápido.

3.3.2 Código de Python

Para empezar la codificación se debe importar las librerías que se encuentran en la Figura 26 necesarias para el desarrollo del proyecto. En la que se incluye la declaración de variables globales que se utilizan en todo el programa Anexo F, en especial para el control del contador.

```
import tkinter
import cv2
import time
import torch
import numpy as np
import tkinter.font as font
from tkinter import ttk
from tkinter import *
import json as json
from PIL import Image, ImageTk
from urllib.request import Request, urlopen
import requests
from tkinter import messagebox
```

```
nueva_imgCam1 = None
historial_etapa1 = 0
historial_etapa2 = 0
historial_etapa3 = 0
historial_etapa4 = 0
historial_etapa5 = 0
```

Figura 26. Importación de librerías y declaración de variables.

Una vez que se ha obtenido el entrenamiento, se sube a donde va a estar el archivo de Python, para posteriormente cargar el modelamiento que va a ser Torch para hacer la conexión a YOLO como se muestra en el código de la Figura 27. Se implementa un retardo de un segundo para permitir que la conexión se establezca adecuadamente. En caso de una conexión deficiente, el sistema está configurado para mostrar un mensaje de error, asegurando así una respuesta adecuada ante posibles problemas de conexión.

```
try:
    time.sleep(1)
    model = torch.hub.load('ultralytics/yolov5', 'custom',
                           path='C:/Users/USER/Documents/Tesis/sistema/Pitahaya1/best.pt')
except:
    print("Error")
```

Figura 27. Conexión a YOLO.

De igual manera se utiliza la función enviar, la cual permite enviar datos a las bases por medio de URL como se visualiza en la Figura 28.

```
def enviar(dir):
    request = Request(dir)
    json = urlopen(request).read().decode()
    return
```

Figura 28. Función enviar.

La función “consultaHistorial” que se muestra en la Figura 29 permite realizar una petición por una URL específica que en este caso es <http://localhost:8080/pitahaya/verAcumulado.php>, para traer los datos del servidor. Los datos recuperados se devuelven en forma de cadena de texto, lo que permite su procesamiento o uso posterior en el ámbito desde el cual se llama a esta función.

```
def consultaHistorial():
    request = Request("http://localhost:8080/pitahaya/verAcumulado.php")
    json = urlopen(request).read().decode()
    return json
```

Figura 29. Realizar una petición por URL.

Luego de ello se emplea la función que permite enviar el historial como se muestra en la Figura 30, para lo cual se inicializa las variables globales, se ubican dichas variables en formato JSON, se envía estos parámetros a la URL y se espera respuesta del servidor.

```

def enviarHistorial():
    global historial_etapa1
    global historial_etapa2
    global historial_etapa3
    global historial_etapa4
    global historial_etapa5
    global etapa1_contador
    global etapa2_contador
    global etapa3_contador
    global etapa4_contador
    global etapa5_contador
    valores_a_enviar = {
        "etapa1": etapa1_contador,
        "etapa2": etapa2_contador,
        "etapa3": etapa3_contador,
        "etapa4": etapa4_contador,
        "etapa5": etapa5_contador
    }
    datos_json = json.dumps(valores_a_enviar)
    try:
        response = requests.post("http://localhost:8080/pitahaya/enviar.php", data=datos_json)
        print("Respuesta desde PHP:", response.text)
        if response.status_code == 200:
            print("Datos enviados correctamente al servidor.")
        else:
            print(f"Error al enviar datos al servidor. Código de estado: {response.status_code}")
    except Exception as e:
        print(f"Error al enviar datos al servidor: {str(e)}")

```

Figura 30 Enviar el historial.

De igual manera, en la Figura 31 se muestra la codificación para la función que permite mostrar el historial. Esta función utiliza la dirección URL establecida para acceder a un recurso en el servidor. Comprueba si la solicitud a la URL ha sido exitosa y ha devuelto datos, esta comprobación es esencial para asegurarse de que hay información válida para mostrar. En caso de que se hayan recibido datos, la función extrae y almacena el valor correspondiente desde la respuesta del servidor. Finalmente, la función utiliza el valor obtenido para actualizar o mostrar la información en los “labels” pertinentes, que son los elementos gráficos en la interfaz gráfica del sistema.

```

def mostrarHistorial():
    global historialGrado1
    global historialGrado2
    global historialGrado3
    global historialGrado4
    global historialGrado5
    global historial_etapa1
    global historial_etapa2
    global historial_etapa3
    global historial_etapa4
    global historial_etapa5

    tDatos = consultaHistorial()
    historial = json.loads(tDatos)

    if historial and len(historial) > 0:
        ultimo_registro = historial[-1]
        historial_etapa1 = ultimo_registro.get('etapa1', 0)
        historial_etapa2 = ultimo_registro.get('etapa2', 0)
        historial_etapa3 = ultimo_registro.get('etapa3', 0)
        historial_etapa4 = ultimo_registro.get('etapa4', 0)
        historial_etapa5 = ultimo_registro.get('etapa5', 0)
        historialGrado1.config(text=f"Grado 1: {historial_etapa1}")
        historialGrado2.config(text=f"Grado 2: {historial_etapa2}")
        historialGrado3.config(text=f"Grado 3: {historial_etapa3}")
        historialGrado4.config(text=f"Grado 4: {historial_etapa4}")
        historialGrado5.config(text=f"Grado 5: {historial_etapa5}")
    else:
        historialGrado1.config(text="Grado 1: 0")
        historialGrado2.config(text="Grado 2: 0")
        historialGrado3.config(text="Grado 3: 0")
        historialGrado4.config(text="Grado 4: 0")
        historialGrado5.config(text="Grado 5: 0")

```

Figura 31. Función mostrar el historial.

La función “capturar” permite conectarse con la cámara, tomar una fotografía, y enviarla al modelo para obtener resultados como se puede observar en la Figura 32. Estos resultados indican en qué etapa se encuentra el proceso de selección de pitahayas.

```
def capturar():
    global lblImgCam1
    global nueva_imgCam1
    global deteccionGrado1
    global deteccionGrado2
    global deteccionGrado3
    global deteccionGrado4
    global deteccionGrado5
    global historial_etapa1
    global historial_etapa2
    global historial_etapa3
    global historial_etapa4
    global historial_etapa5
    global historialGrado1
    global historialGrado2
    global historialGrado3
    global historialGrado4
    global historialGrado5
    global etapa1_contador
    global etapa2_contador
    global etapa3_contador
    global etapa4_contador
    global etapa5_contador
    cap = cv2.VideoCapture(0)

    mostrarHistorial()

    leido0, frame0 = cap.read()
    cap.release()
    cv2.imwrite("camara.jpg", frame0)
    image_path = "camara.jpg"
    #frame1 = cv2.imread(image_path)
    im1 = Image.open(image_path)
    model.conf = 0.80
    print("pasos")
```

Figura 32. Función capturar.

En la Etapa 1, se realiza un tratamiento de imágenes para determinar el porcentaje de color verde presente en la fotografía capturada. Esto sirve para verificar si el porcentaje es mayor, lo cual indica que la pitahaya está en grado 2; de lo contrario, se concluye que está en grado 1. En la Etapa2, se almacena la pitahaya como grado 3. En la Etapa 3, se efectúa un tratamiento de imágenes para evaluar el porcentaje de color amarillo presente. Esto determina si el porcentaje es mayor, colocando la pitahaya en grado 5; en caso contrario, se clasifica como grado 4. Dentro de cada comprobación se procede a realizar el conteo de pitahayas en cada grado y estos resultados se muestran. Al finalizar este proceso se suma estas cantidades al historial actual como se muestra la Figura 33.

```

for i in range(len(info)):
    if len(info.iloc[i]) >= 6:
        clase = int(info.iloc[i]['class'])

        if clase == 0:
            print(clase)
            objeto = im1.crop((info.iloc[i]['xmin'], info.iloc[i]['ymin'], info.iloc[i]['xmax'], info.iloc[i]['ymax']))
            porcentaje_verde = tratamiento_etapa1(objeto)
            if porcentaje_verde > 30:
                etapa2_contador += 1
            else:
                etapa1_contador += 1
        elif clase == 1:
            print(clase)
            etapa3_contador += 1
        elif clase == 2:
            print(clase)
            objeto = im1.crop((info.iloc[i]['xmin'], info.iloc[i]['ymin'], info.iloc[i]['xmax'], info.iloc[i]['ymax']))
            porcentaje_amarillo, porcentaje_verde = tratamiento_etapa3(objeto)
            print(porcentaje_amarillo)
            if porcentaje_amarillo > 44:
                etapa5_contador += 1
            else:
                etapa4_contador += 1

```

Figura 33. Tratamiento de imágenes para determinar el porcentaje de color verde.

Como se observa en la Figura 34, a continuación, se emplea la función “tratamiento_etapa1”, la cual permite verificar o controlar la etapa1, es decir, este código realiza un procesamiento de las imágenes para calcular y retornar el porcentaje de área verde en la imagen de la pitahaya analizada.

```

def tratamiento_etapa1(objeto):
    img_cv2 = np.array(objeto)
    img_cv2 = cv2.cvtColor(img_cv2, cv2.COLOR_RGB2BGR)
    img_hsv = cv2.cvtColor(img_cv2, cv2.COLOR_BGR2HSV)
    lower_green = np.array([40, 40, 40])
    upper_green = np.array([80, 255, 255])
    mask_green = cv2.inRange(img_hsv, lower_green, upper_green)
    area_total = objeto.size[0] * objeto.size[1]
    area_verde = cv2.countNonZero(mask_green)
    porcentaje_verde = (area_verde / area_total) * 100
    print(f"Porcentaje de Verde: {porcentaje_verde:.2f}%")

    return porcentaje_verde

```

Figura 34. Controlar la etapa1.

En la Figura 35 se realiza la codificación para el tratamiento de la etapa3 de maduración de la pitahaya, para lo cual se empleó la función “tratamiento_etapa3”.

```

def tratamiento_etapa3(objeto):
    img_cv2 = np.array(objeto)
    img_cv2 = cv2.cvtColor(img_cv2, cv2.COLOR_RGB2BGR)
    img_hsv = cv2.cvtColor(img_cv2, cv2.COLOR_BGR2HSV)
    lower_yellow = np.array([20, 100, 100])
    upper_yellow = np.array([30, 255, 255])
    lower_green = np.array([40, 40, 40])
    upper_green = np.array([80, 255, 255])
    lower_brown = np.array([10, 60, 20])
    upper_brown = np.array([20, 255, 255])
    mask_yellow = cv2.inRange(img_hsv, lower_yellow, upper_yellow)
    mask_green = cv2.inRange(img_hsv, lower_green, upper_green)
    mask_brown = cv2.inRange(img_hsv, lower_brown, upper_brown)
    area_total = objeto.size[0] * objeto.size[1]
    area_amarillo = cv2.countNonZero(mask_yellow)
    area_verde = cv2.countNonZero(mask_green)
    area_cafe = cv2.countNonZero(mask_brown)
    porcentaje_amarillo = (area_amarillo / area_total) * 100
    porcentaje_verde = (area_verde / area_total) * 100
    porcentaje_cafe = (area_cafe / area_total) * 100
    print(f"Porcentaje de Amarillo: {porcentaje_amarillo:.2f}%")
    print(f"Porcentaje de Verde: {porcentaje_verde:.2f}%")
    print(f"Porcentaje de Café: {porcentaje_cafe:.2f}%")
    img_cafe = cv2.bitwise_and(img_cv2, img_cv2, mask=mask_brown)
    img_cafe_pil = Image.fromarray(cv2.cvtColor(img_cafe, cv2.COLOR_BGR2RGB))
    img_cafe_pil.save("proceso3.png")

    return porcentaje_amarillo, porcentaje_verde

```

Figura 35. Codificación para el tratamiento de la etapa3.

3.3.3 Base de Datos

La base de datos realizada se efectuó una prueba inicial en un servidor local utilizando XAMPP. Luego, se accede a la configuración de MySQL a través de PHPMyAdmin, la cual es una interfaz web que permite administrar bases de datos MySQL. En el proceso, se crea una base de datos denominada “pitahaya” como indica la Figura 36. Dentro de esta base de datos, se establece una tabla con el nombre de “historial”. Esta tabla de la base de datos está diseñada para almacenar información del historial relacionada con la selección de cosecha en el cultivo de pitahaya. La creación de esta tabla sirve como repositorio estructurado, para almacenar datos importantes generados por el sistema inteligente.

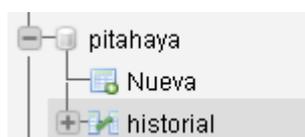


Figura 36. Base de datos

La tabla de datos consta de siete campos que son: id, etapa1, etapa2, etapa3, etapa4, etapa5 y fecha como se muestra en la Figura 37.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/> 2	etapa1	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	etapa2	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 4	etapa3	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 5	etapa4	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 6	etapa5	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 7	fecha	datetime			No	current_timestamp()			Cambiar Eliminar Más

Figura 37. Tabla de datos de selección de la pitahaya.

3.3.4 Modelo de la tabla

El modelo de tabla diseñada para la base de datos del sistema inteligente de selección de cosecha en el cultivo de pitahaya ofrece una estructura como indica la Figura 38 la cual permite acumular de manera efectiva el historial de lecturas correspondientes a cada grado de madurez de la pitahaya. La inclusión de varios registros en la base de datos facilita la recopilación y organización sistemática de la información, lo que resulta esencial para el análisis y seguimiento preciso del desarrollo y calidad de la fruta en el cultivo.



#	Nombre	Tipo
#	etapa1	int(11)
#	etapa2	int(11)
#	etapa3	int(11)
#	etapa4	int(11)
#	etapa5	int(11)

Figura 38. Registros en la base de datos.

En cuanto a los archivos PHP que posibilitan la interacción con la base de datos, se detallan en la Figura 39.



Figura 39. Archivos PHP.

- **Acceso.php:** En este código se ubica los parámetros para la conexión a la base de datos. Contiene la información como el nombre de usuario, contraseña, host y nombre de la base de datos, que se requiere para establecer una conexión efectiva entre el sistema y la base de datos.
- **Enviar.php:** Este archivo permite realizar la inserción del historial de los grados de la pitahaya, recibiendo como parámetro cada valor. En la sección superior, invoca los archivos de conexión y permisos de acceso desde un URL. Es fundamental para almacenar de manera efectiva los datos recopilados durante el proceso de selección de cosecha.
- **Permiso.php:** Código para dar acceso a diferentes URL y método de recepción de datos. Establece las condiciones y permisos necesarios para la interacción controlada con la base de datos, garantizando la seguridad y la integridad de los datos almacenados.
- **verAcumulado.php:** Permite seleccionar de la base de datos la suma de los valores de cada grado de pitahaya. Facilita la obtención de datos acumulados, lo que resulta valioso para evaluar tendencias y patrones en el desarrollo de la fruta a lo largo del tiempo.

3.3.5 Interfaz Gráfica del Sistema de Control de Calidad de la Pitahaya

La interfaz del sistema permite al usuario identificar los grados de maduración de la pitahaya para exportación y consumo nacional de manera intuitiva y fácil de manipular. En la pantalla principal se muestra el ingreso al sistema mediante en la cual solicita los datos del usuario junto con la contraseña a su vez un botón de ingreso el cual permite ingresar al sistema del control de calidad con el fin de clasificar por los grados de maduración de la pitahaya e ir acumulando los resultados de las pruebas como se observa en la Figura 40.

Además, como se muestra en la

Figura 41, se cuenta con una ventana que presenta la detección actual de las frutas, es decir el número de pitahayas que se encuentran en la caja y la respectiva etapa a la que pertenece, y una segunda columna que visualiza el historial de detección que se ha realizado desde que se ejecutó el programa.



Figura 40. Interfaz Gráfica del Sistema de Control de Calidad de la Pitahaya.

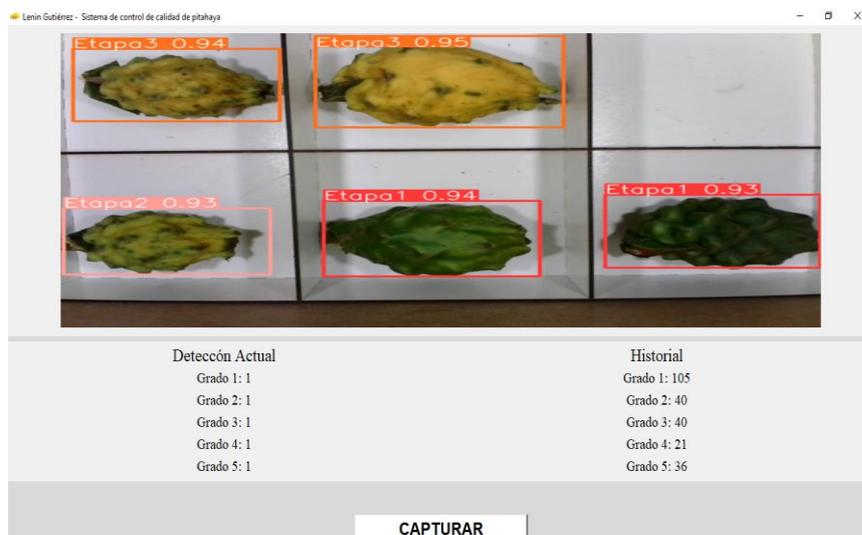


Figura 41 Detección actual de las frutas con una columna con el número de pitahayas y una segunda columna con el historial de las pitahayas según sus grados

a. Diseño del Prototipo

El prototipo diseñado consta de dos partes la gaveta y la cabina para el sistema de visión artificial. Por parte de la gaveta se diseñó con el fin de analizar las pitahayas de forma múltiple mas no individual por lo tanto esta gaveta realizada en MDF consta de 6 compartimientos como se indica la Figura 42, el proceso de captura de la imagen se utilizó la cámara Trust GXT1160, la cual fue ubicada estratégicamente en el entorno para abarcar la mayor superficie posible y obtener imágenes representativas de las pitahayas. La iluminación y el fondo son parámetros esenciales que se tuvieron que considerar para obtener imágenes nítidas y de esa manera garantizar el adecuado funcionamiento del sistema.

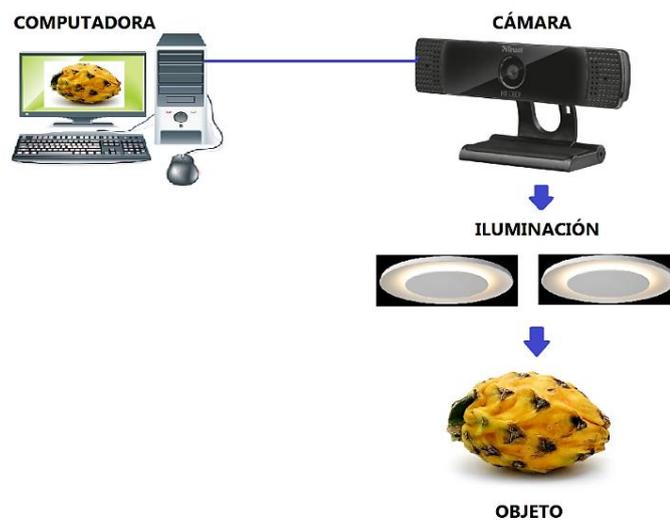


Figura 42. Esquema de la adquisición de la imagen.

El sistema de adquisición de imágenes, Anexo E está compuesto por una cabina que alberga una cámara web, dispuesta a una distancia de 55 cm desde la base donde se posiciona el objeto de interés. Como se observa en la Figura 43, esta cabina presenta un fondo blanco homogéneo que previene la aparición de sombras o reflejos no deseados. Para optimizar la iluminación, se incorporaron dos luminarias LED estratégicamente colocadas en la parte superior de la cabina.

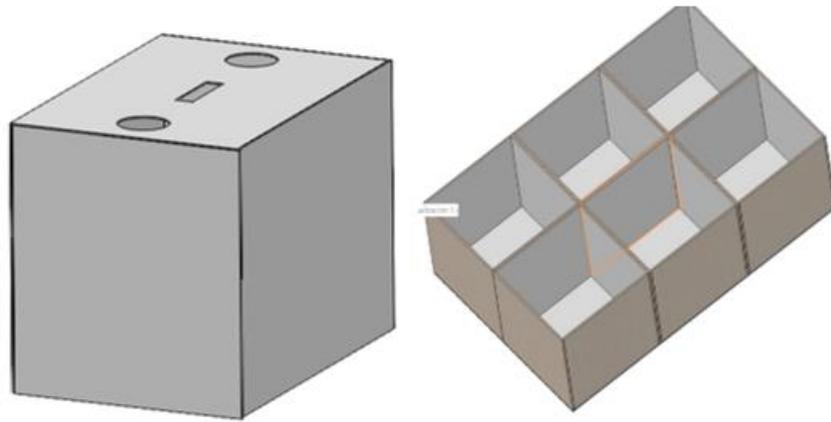


Figura 43. Cabina de visión artificial.

Una vez realizada el diseño se lo implemento Anexo G con la cámara web seleccionada al igual de la luz led artificial como se observa en la siguiente Figura 44.



Figura 44. Implementación del prototipo.

3.3.6 Diagrama de flujo del sistema

En el diagrama de flujo del sistema que se muestra en la Figura 45, se observa el proceso de funcionamiento del sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial. El algoritmo diseñado refleja la integración y funcionalidad del sistema que en primera instancia solicita ingreso los datos del

usuario, si son incorrectos le presenta un mensaje de error, si son correctos pasa a la ventana principal.

En la ventana principal muestra una sección superior y horizontal la misma donde se muestra la captura de la imagen obtenida desde la cámara. Una vez que se tiene esta foto, el sistema comprueba con el modelo entrenado en Yolov5, Se entrenó con tres etapas, el grado 1 y 2, aparecen en la etapa 1, el grado 3 aparece como etapa2, y el grado 4 y 5 aparece como etapa 3, eso se hizo porque la etapa 1 y 2 son muy similares, así mismo la etapa 4 y 5.

Entonces ese control se hace por tratamiento de imagen para verificar cual tiene más porcentaje de verde o amarillo según el caso dentro de una imagen. Con lo anterior después del obtener el resultado del modelo y verificar cada etapa y haber realizado este proceso de agrupación en grados, se cuenta cuántas pitahayas de cada grado hay en la imagen y se muestra ese valor, después esos valores se suman al historial global.

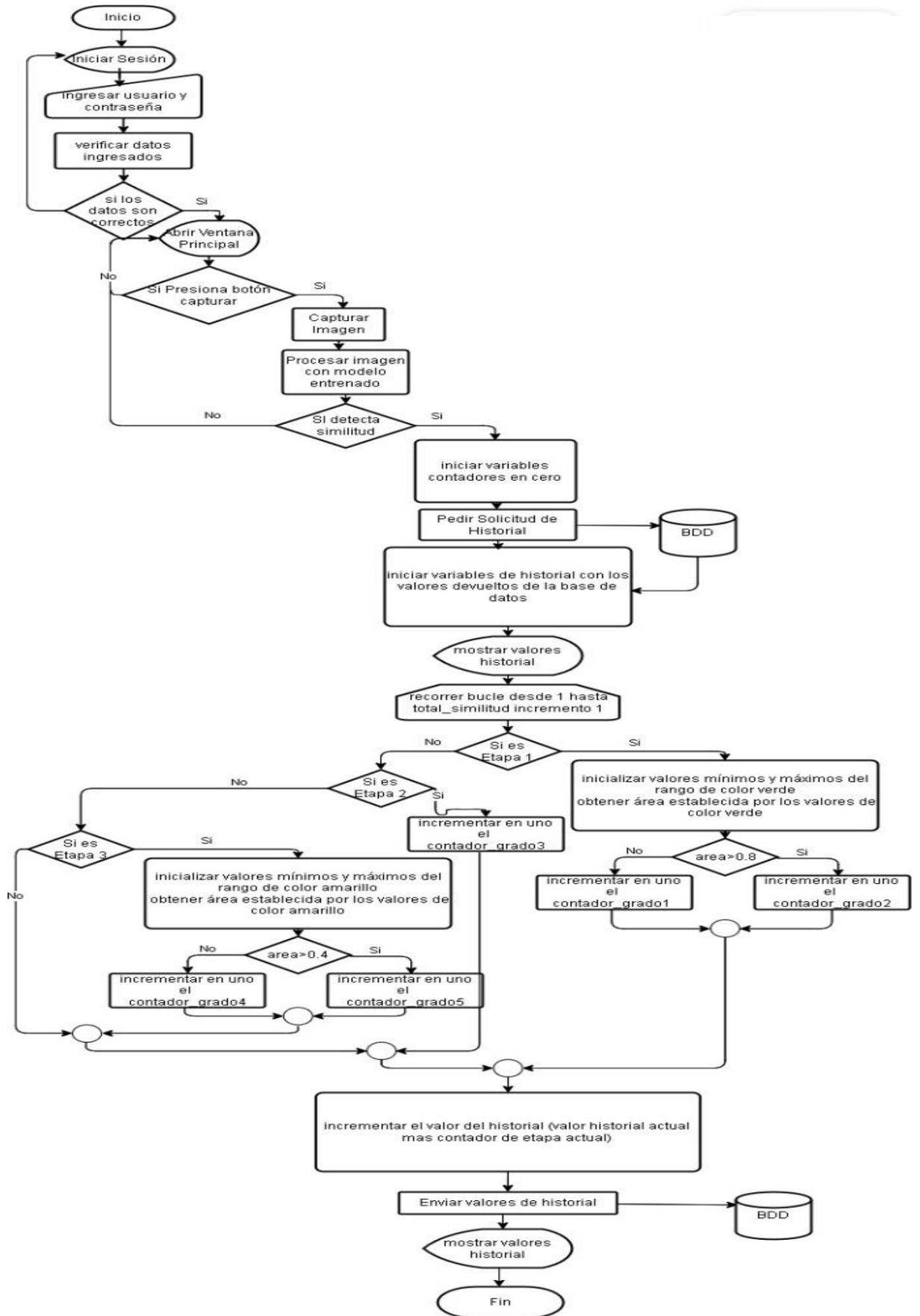


Figura 45. Diagrama de flujo del sistema.

3.4 Pruebas del entrenamiento YOLOv5

Una vez que se finaliza el entrenamiento de la red, se verificó el desempeño del mismo por medio de la validación de algoritmos con curvas de presión, sensibilidad, matriz de confusión, por medio de la herramienta Clearml, la cual es integrada de manera automática al entrenamiento de YOLOv5.

3.4.1 Distribución de clases

Una vez realizado el entrenamiento del modelo con 200 épocas, con el set de datos reducidos a las categorías necesarias y con imágenes que posee buena resolución e iluminación, se observa los resultados de la Figura 46. Se visualiza que la distribución de las diversas etapas de las pitahayas es algo desigual; no obstante, se aprecia una cantidad suficiente de regiones delimitadas (boxes) para cada categoría. Este hallazgo sugiere que, a pesar de la disparidad en la distribución, el conjunto de datos proporciona una variedad adecuada de instancias de cada categoría. Este factor es esencial para garantizar que el proceso de entrenamiento sea exitoso y que el modelo pueda aprender de manera efectiva las características distintivas de cada etapa de las pitahayas.

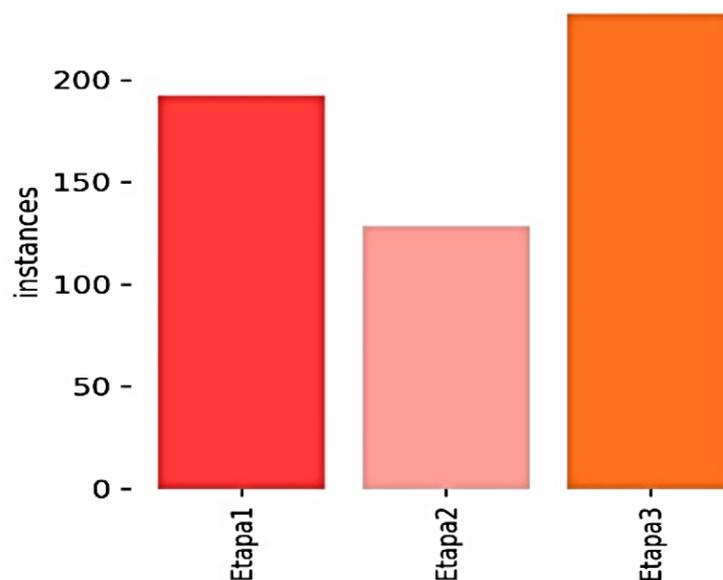


Figura 46. Distribución de clases.

3.4.2 Validación de resultados

En el sistema desarrollado, se llevaron a cabo varios entrenamientos con el objetivo de realizar una comparación de resultados. En las tablas siguientes se presentan los valores obtenidos de los entrenamientos, que incluyen los parámetros de P (Precisión), R (Recall), mAP50 (mean Average Precision at IoU 0.5) y mAP50-95 (mean Average Precision entre IoU 0.5 y 0.95). En la Tabla 16 muestra los resultados obtenidos después de completar el entrenamiento con 135 épocas:

Tabla 16. Resultado de 135 épocas.

Clases	Imágenes	Instancias	P	R	mAP50	mAP50-95
All	135	135	0,754	0,947	0,968	0,931
Etapa 1	135	27	1	0,885	0,984	0,954
Etapa 2	135	27	0,496	1	0,966	0,963
Etapa 3	135	27	0,969	1	0,995	0,949
Etapa 4	135	27	0,752	0,852	0,945	0,899
Etapa 5	135	27	0,551	1	0,951	0,892

Posteriormente, se llevó a cabo otro entrenamiento con 200 épocas para obtener resultados más confiables, como se muestra en la Tabla 17.

a. Matriz de confusión

La matriz de confusión, empleada en la evaluación del rendimiento del algoritmo YOLOv5, se genera de manera individual para cada etapa y conjunto de datos recolectados. Ofrece detalles porcentuales sobre la exactitud de características y criterios específicos. Esta matriz refleja las detecciones del modelo para cada etapa durante la fase de validación cada valor en la diagonal principal representa el número de instancias correctamente clasificadas para cada etapa, como indica la Figura 47, todos los valores fuera de la diagonal principal son cero, lo cual indica que no hay errores de clasificación. Cada etapa tiene una correspondencia perfecta entre las instancias reales y las predichas. En resumen, esta matriz de confusión refleja un modelo que clasifica todas las instancias en las tres etapas correctamente, sin errores de clasificación.

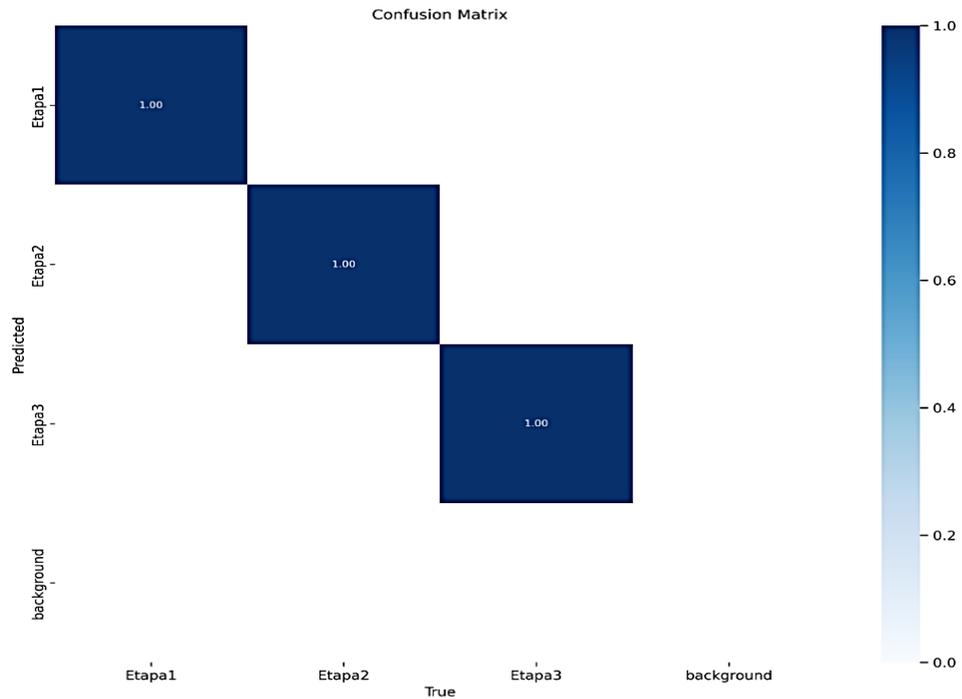


Figura 47. Matriz de confusión.

La Tabla 17, presenta los resultados obtenidos después de completar el entrenamiento con 200 épocas:

Tabla 17. Resultado de 200 épocas.

Clases	Imágenes	Instancias	P	R	mAP50	mAP50-95
All	160	160	0,999	1	0,995	0,936
Etapa1	160	66	0,999	1	0,995	0,978
Etapa2	160	37	0,999	1	0,995	0,988
Etapa3	160	57	0,999	1	0,995	0,923

Para la comparación de estas métricas mencionadas en las tablas Tabla 16 y Tabla 17 se selecciona la fila correspondiente a 'All', la cual presenta los valores medidos de cada entrenamiento.

- Precisión:** La precisión es una medida que evalúa la proporción de predicciones positivas correctas en relación con todas las predicciones positivas realizadas por el modelo. En el primer entrenamiento, se obtuvo un valor de precisión de 0,754, mientras que en el último entrenamiento se logró una precisión de 0,999. Esto indica que el modelo tiene una capacidad muy alta para realizar predicciones precisas, con una tasa mínima de falsos positivos.

- **Recall:** El recall, también conocido como sensibilidad o tasa de verdaderos positivos, mide la proporción de instancias positivas que el modelo ha identificado correctamente en comparación con el total de instancias positivas en el conjunto de datos. En el entrenamiento inicial, se alcanzó un valor de recall de 0,947, mientras que, en el entrenamiento final, el recall alcanza 1. Esto sugiere que el modelo ha identificado todas las instancias positivas, no dejando ningún positivo sin detectar.
- **mAP50 (mean Average Precision at IoU 0.5):** Esta métrica indica la calidad general de las predicciones del modelo, evaluando la precisión promedio para diferentes umbrales de IoU (Intersección sobre Unión) con un valor específico de 0,5. En el entrenamiento inicial, arroja un valor de 0,968, mientras que el entrenamiento final registra una precisión de 0,995. Esto sugiere que las predicciones del modelo son altamente precisas en términos de solapamiento con las instancias reales
- **mAP50-95 (mean Average Precision entre IoU 0.5 y 0.95):** mAP50-95 mide la precisión promedio para un rango más amplio de umbrales de IoU desde 0.5 hasta 0,95 en la sesión de entrenamiento inicial, se obtuvo un valor de 0,931, mientras que, en la sesión de entrenamiento final se registró un valor de 0,936. Esto sugiere una buena consistencia en la precisión del modelo a lo largo de este rango de umbrales de IoU.

En conjunto, estos resultados indican que el modelo ha logrado un rendimiento eficaz en términos de precisión y recuperación, con una alta capacidad para realizar predicciones precisas y recuperar correctamente las instancias positivas en el conjunto de datos de prueba.

3.4.3 Indicador F1

El indicador F1, también conocido como la puntuación F1 o medida F1, es una métrica que combina la precisión y el recall en una sola medida, proporcionando una evaluación más equilibrada del rendimiento de un modelo. Los resultados de la Figura 48 indican un equilibrio entre la capacidad del modelo para hacer predicciones precisas

y su capacidad para capturar todas las instancias positivas en el conjunto de datos, debido a que alcanza un intervalo $0,9 \geq 1$, lo cual sugiere un resultado favorable.

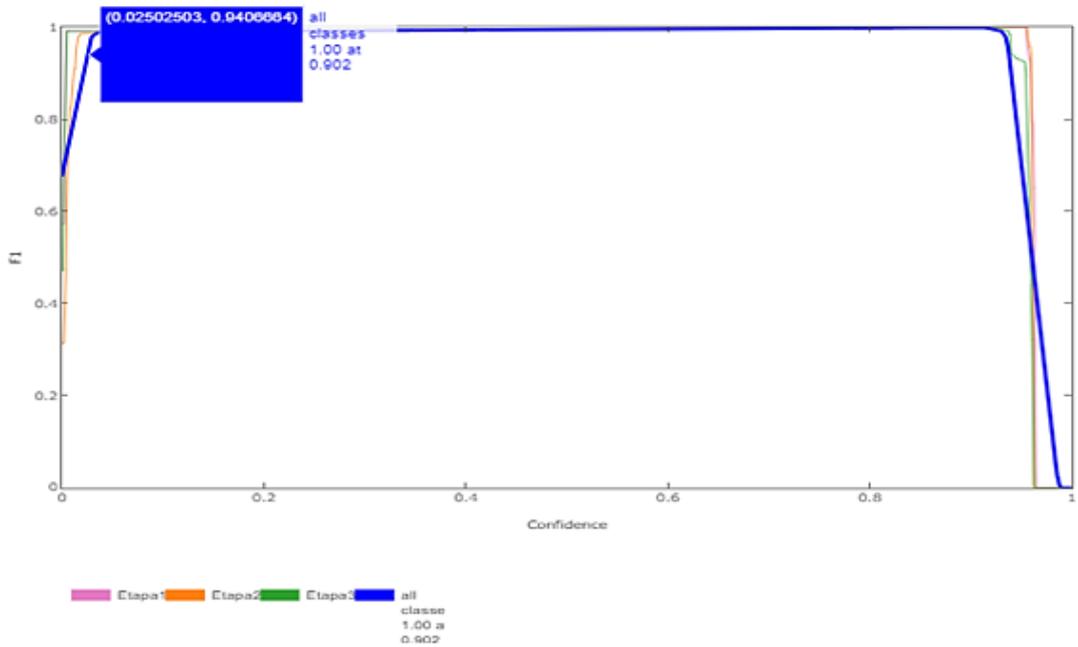


Figura 48. Indicador F1.

3.4.4 Imágenes del testeo de clasificación de la pitahaya

La evaluación que se realiza para la determinación de la etapa de maduración de la pitahaya, cuyo modelo fue realizado en 200 épocas presentaron una precisión del 99,9 %. En la Figura 49 que se muestra a continuación se presentan ejemplos del reconocimiento de la fruta en cada una de las etapas con su respectivo porcentaje de precisión.

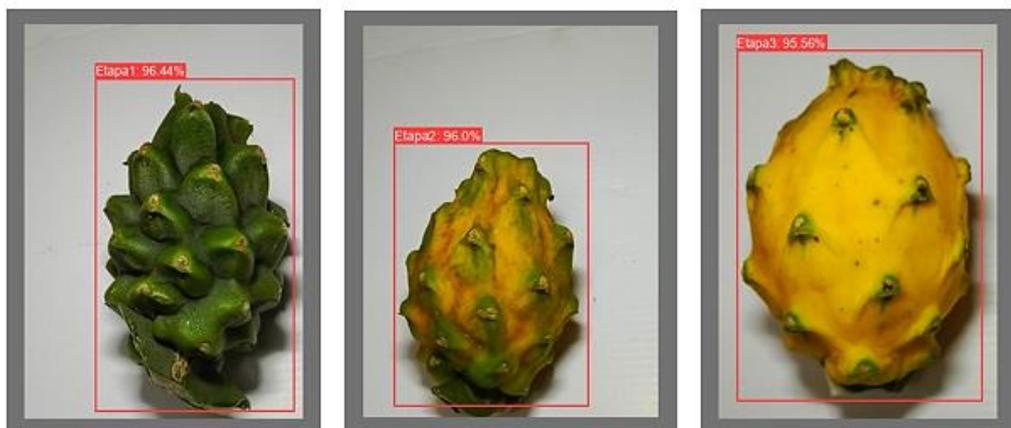


Figura 49. Imágenes del testeo de clasificación de la pitahaya.

3.4.5 Pruebas de funcionamiento del sistema

a. Pruebas en tiempo real de cada una de las etapas de maduración de la pitahaya

En la Figura 50. se puede observar la disposición del prototipo implementado del sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial, conformado por una computadora junto con la cabina de visión artificial.

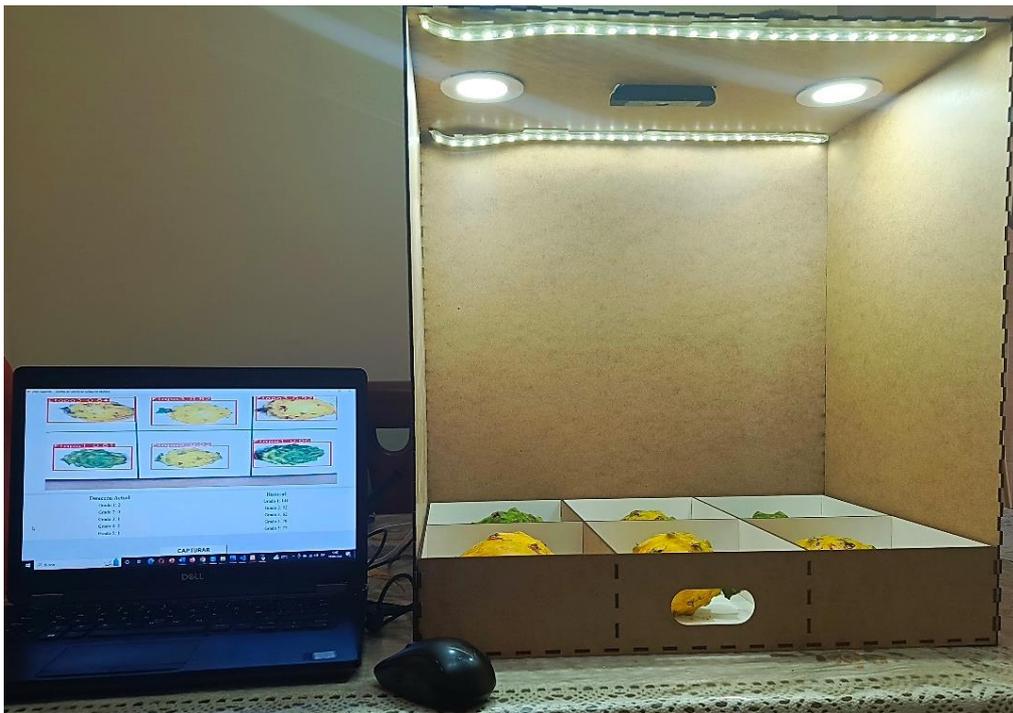
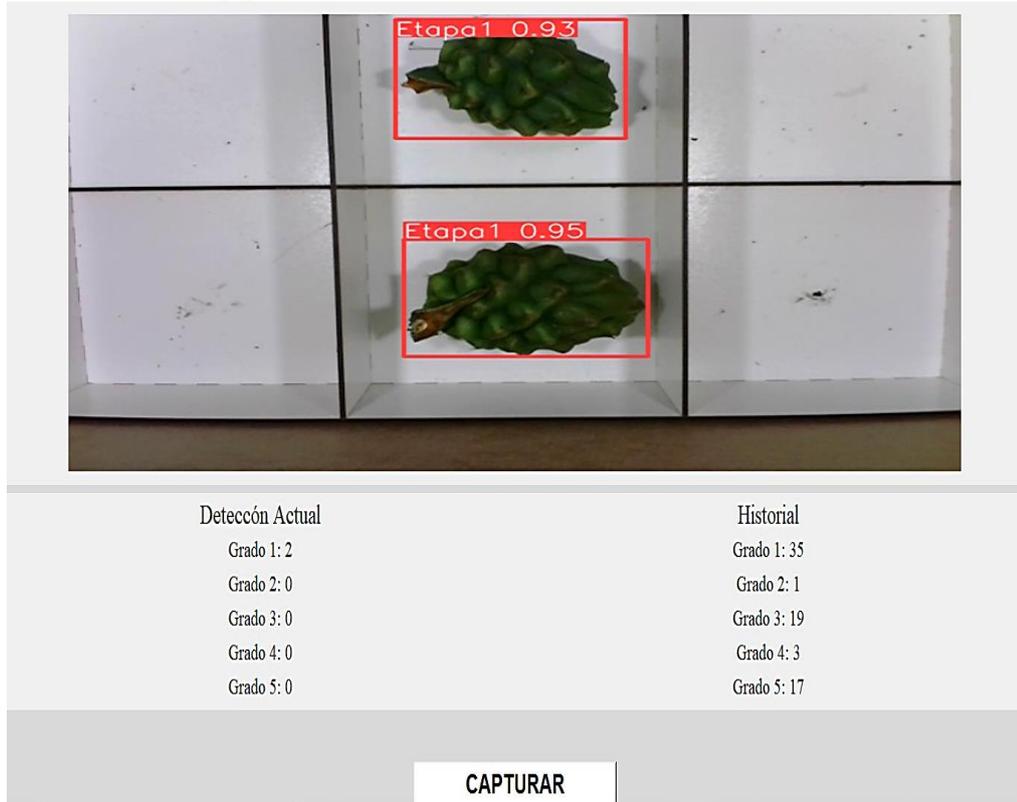


Figura 50. Prototipo implementado del sistema de sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial

En las figuras (Figura 51, Figura 52, Figura 53) que se muestran a continuación se puede evidenciar por capturas de las imágenes de la pitahaya en tiempo real con el respectivo reconocimiento de la etapa que reconoce y el porcentaje de precisión respectivo.

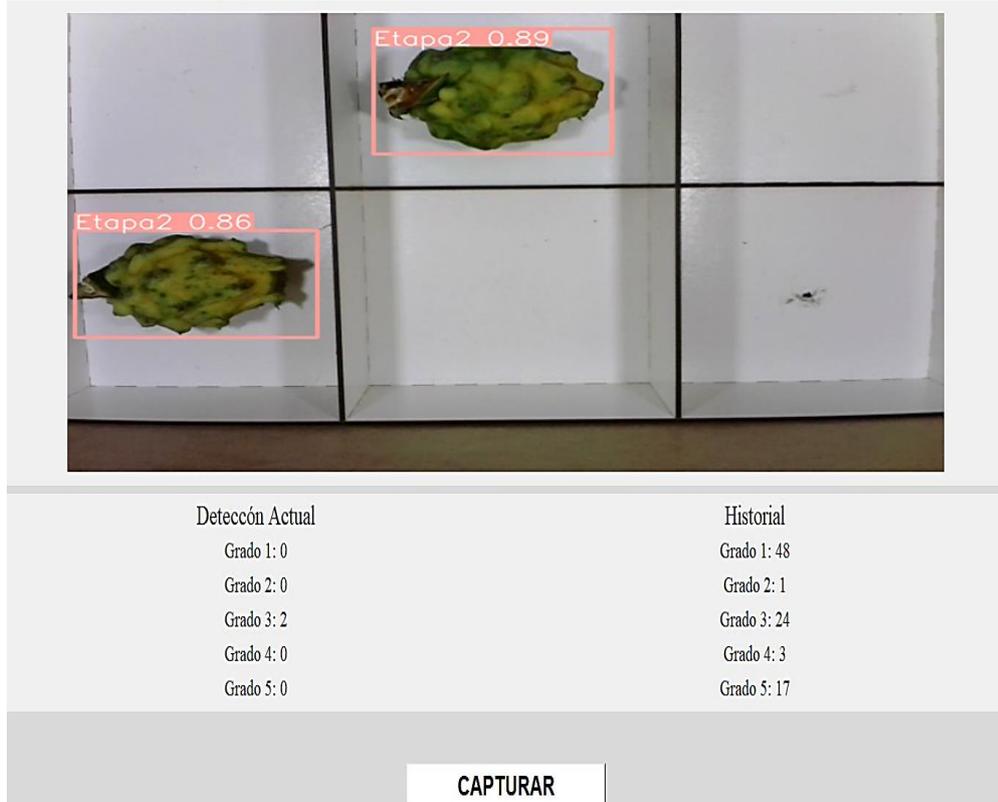


The screenshot shows a software interface for pitahaya quality control. It features a 2x2 grid of images. The top-right image shows a green pitahaya with a red bounding box and the text "Etapa1 0.93". The bottom-left image shows a green pitahaya with a red bounding box and the text "Etapa1 0.95". Below the grid, there are two columns of data: "Detección Actual" and "Historial".

Detección Actual	Historial
Grado 1: 2	Grado 1: 35
Grado 2: 0	Grado 2: 1
Grado 3: 0	Grado 3: 19
Grado 4: 0	Grado 4: 3
Grado 5: 0	Grado 5: 17

CAPTURAR

Figura 51. Pruebas de funcionamiento pitahaya etapa 1.



The screenshot shows a software interface for pitahaya quality control. It features a 2x2 grid of images. The top-right image shows a green pitahaya with a red bounding box and the text "Etapa2 0.89". The bottom-left image shows a green pitahaya with a red bounding box and the text "Etapa2 0.86". Below the grid, there are two columns of data: "Detección Actual" and "Historial".

Detección Actual	Historial
Grado 1: 0	Grado 1: 48
Grado 2: 0	Grado 2: 1
Grado 3: 2	Grado 3: 24
Grado 4: 0	Grado 4: 3
Grado 5: 0	Grado 5: 17

CAPTURAR

Figura 52. Pruebas de funcionamiento pitahaya etapa 2.

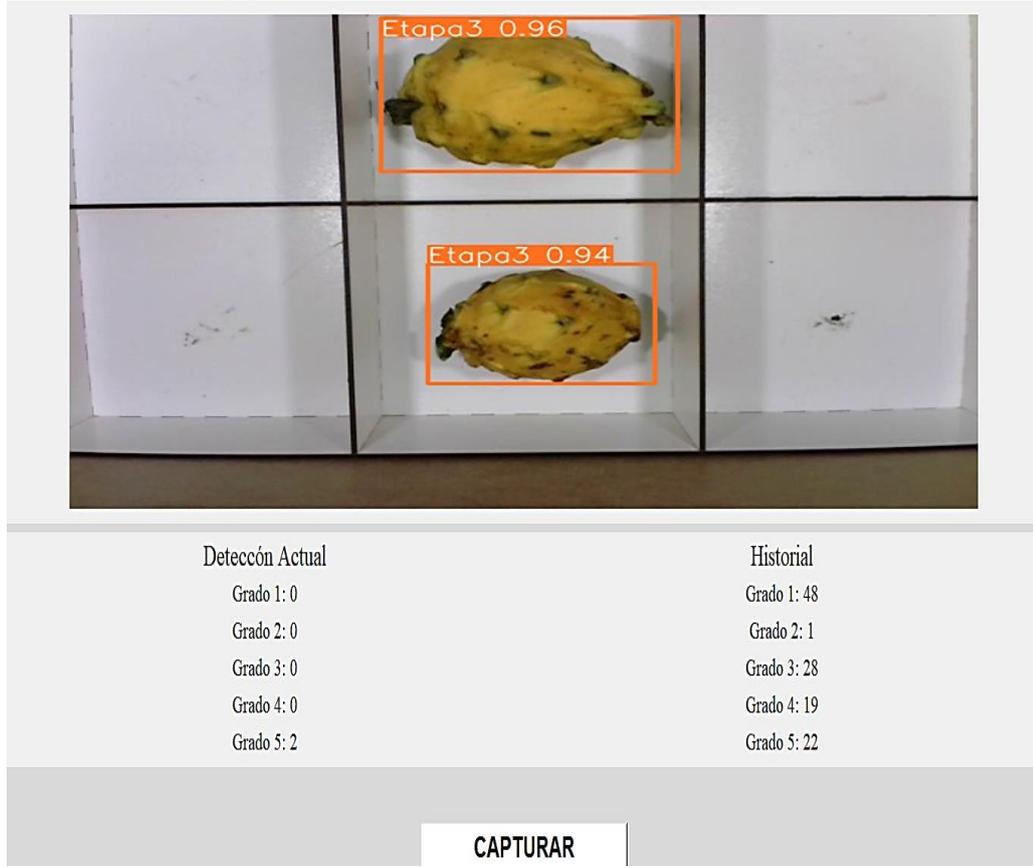


Figura 53. Pruebas de funcionamiento pitahaya etapa 3.

b. Pruebas realizadas del funcionamiento del sistema de visión artificial

En la Tabla 18 y Figura 54 se indica los valores obtenidos de las pruebas realizadas del sistema de visión artificial para determinar el grado de maduración de la pitahaya. Estas pruebas indica el número de pitahayas que se coloca en la gaveta, se tiene una inspección visual humana y una inspección artificial por lo tanto se obtiene para grado 1: 89,65%, grado 2: 88,56%, grado 3: 90,09%, grado 4: 86,93%, grado 5: 89,21%. El tiempo promedio de captura de la imagen hasta su clasificación fue de 4,48 s.



Figura 54. Pruebas de funcionamiento del sistema

Tabla 18. Pruebas de funcionamiento del sistema.

# PRUEBAS	# PITAHAYAS	INSPECCIÓN HUMANA VISUAL					INSPECCIÓN ARTIFICIAL					
		GRADOS DE MADURACIÓN					TIEMPO	GRADOS DE MADURACIÓN				
		1	2	3	4	5		1	2	3	4	5
PRUEBA 1	6							2		1	1	2
		1	1	1	1	2	4,32	0,82		0,90	0,91	0,93
PRUEBA 2	3							2	1			
		2	1	0	0	0	4,63	0,88	0,89			
PRUEBA 3	4							3	1			
		2	2	0	0	0	4,41	0,89	0,91			
PRUEBA 4	4							2			1	1
		1	1	0	1	1	4,34	0,81			0,92	0,92
PRUEBA 5	3							1			1	1
		1	0	0	1	1	4,45	0,80			0,91	0,91
PRUEBA 6	5								1	2	1	1
		0	1	2	1	1	4,48		0,88	0,87	0,85	0,94
PRUEBA 7	6							1	1	2	1	1
		1	1	2	1	1	4,33	0,91	0,88	0,90	0,89	0,95
PRUEBA 8	4							1	1	1	1	
		1	1	1	1	0	4,16	0,88	0,89	0,86	0,90	
PRUEBA 9	4							1		2	1	
		1	0	2	1	0	4,47	0,91		0,87	0,94	
PRUEBA 10	3									2	1	
		0	0	2	1	0	4,48			0,88	0,94	
PRUEBA 11	5								2	2	1	
		1	1	2	1	0	4,54					

# PRUEBAS	# PITAHAYAS	INSPECCIÓN HUMANA VISUAL					INSPECCIÓN ARTIFICIAL						
		GRADOS DE MADURACIÓN					TIEMPO	GRADOS DE MADURACIÓN					
		1	2	3	4	5		1	2	3	4	5	
PRUEBA 12	4	0	1	1	1	1	4,15		0,91	0,90	0,89		
								1	1	2			
									0,87	0,89	0,94		
PRUEBA 13	4	0	1	1	2	0	4,56		1	1	2		
									0,90	0,91	0,93		
PRUEBA 14	3	0	0	1	0	2	4,34			1		2	
										0,90		0,94	
PRUEBA 15	3	1	0	0	1	1	4,32	1			1	1	
								0,90			0,93	0,92	
PRUEBA 16	2	1	1	0	0	0	4,33	1	1				
								0,95	0,89				
PRUEBA 17	6	1	1	3	0	1	4,3	1	1	2	2		
								0,90	0,93	0,90	0,88		
PRUEBA 18	6	2	0	2	0	2	4,19	1	1	2		2	
								0,88	0,85	0,90		0,93	
PRUEBA 19	6	0	3	3	0	0	4,34	2	1	3			
								0,90	0,85	0,92			
PRUEBA 20	6	1	1	1	1	1	4,49	1	1	1	1	1	
								0,94	0,93	0,92	0,95	0,96	
PRUEBA 21	6	1	2	0	2	1	4,52	1	2		1	2	
								0,88	0,90		0,87	0,86	
PRUEBA 22	6	3	1	2	0	0	4,39	3	1	2			
								0,90	0,90	0,94			
PRUEBA 23	4	2	2	0	0	0	4,93	3	1				
								0,90	0,89				
PRUEBA 24	5	1	1	2	1	0	4,36	1	1	2		1	
								0,92	0,96	0,93		0,90	
PRUEBA 25	6	2	0	3	0	1	4,4	2		3		1	
								0,85		0,89		0,94	
PRUEBA 26	6	2	1	1	2	0	4,2	1	2	1	1	1	
								0,85	0,90	0,85	0,87	0,91	
PRUEBA 27	5	1	2	1	1	0	4,27	2	1	1	1		
								0,85	0,91	0,86	0,89		
PRUEBA 28	4	0	2	0	2	0	4,49		2		1	1	
									0,84		0,90	0,00	
PRUEBA 29	3	1	0	0	0	2	4,35	1				2	
								0,87				0,88	
PRUEBA 30	4	1	1	1	0	1	4,6	1	1	1		1	
								0,84	0,88	0,88		0,92	
PRUEBA 31	5	1	1	2	1		4,18	1	1	2	1		
								0,89	0,82	0,87	0,88		

# PRUEBAS	# PITAHAYAS	INSPECCIÓN HUMANA VISUAL					INSPECCIÓN ARTIFICIAL						
		GRADOS DE MADURACIÓN					TIEMPO	GRADOS DE MADURACIÓN					
		1	2	3	4	5		1	2	3	4	5	
PRUEBA 32	2	0	1	0	0	1	4,26		1			1	
									0,88			0,94	
PRUEBA 33	4	1	1	0	1	1	4,49	1	1		2		
								0,92	0,88		0,04		
PRUEBA 34	3	0	2	0	0	1	4,38	1	1			1	
								0,94	0,94			0,87	
PRUEBA 35	4	2	0	2	0	0	4,77	2		2			
								0,94		0,91			
PRUEBA 36	5	2	2	0	0	1	4,93	3	1			1	
								0,93	0,91	0,92		0,96	
PRUEBA 37	6	3	2	1	0	0	4,43	3	2	1			
								0,93	0,88	0,92			
PRUEBA 38	6	1	2	1	1	1	4,23	1	2	1	1	1	
								0,92	0,82	0,92	0,92	0,95	
PRUEBA 39	5	1	0	3	0	1	4,48	1		3		1	
								0,93		0,92		93%	
PRUEBA 40	4	0	0	3	1	0	4,35			3	1		
										0,93	0,93	0,95	
PRUEBA 41	4	0	0	1	2	2	4,46			1	1	3	
										0,85	0,94	0,95	
PRUEBA 42	6	0	1	2	1	2	5,95		1	2		3	
									0,88	0,93		0,95	
PRUEBA 43	2	0	2	0	0	0	4,73		2				
									0,83				
PRUEBA 44	5	1	1	0	1	2	4,24	1	1		1	2	
								0,95	0,87		0,90	0,96	
PRUEBA 45	1	0	0	0	0	1	5,28					1	
												0,91	
PRUEBA 46	6	1	2	2	0	1	5,31	1	2	2		1	
								0,95	0,87	0,92		0,92	
PRUEBA 47	4	2	0	2	0	0	4,63	2		2			
								0,89		0,95			
PRUEBA 48	6	1	2	2	1	0	4,48	1	2	2	1		
								0,91	0,88	0,93	0,84		
PRUEBA 49	4	1	0	1	1	1	4,35	1		1	1	1	
								0,92		0,89	0,83	0,89	
PRUEBA 50	5	1	1	1	1	1	4,37	1	1	1	1	1	
								0,92	0,89	0,90	0,88	0,88	
PROMEDIO							4,49		0,90	0,89	0,90	0,87	0,89
PROMEDIO (%)									89,7	88,56	90,09	86,93	89,21

c. Cálculo del error

El cálculo del error se lo realizo mediante las pitahayas colocadas en el prototipo en cada prueba realizada y las que detecto correctamente el sistema para ello se emplean los valores para cada grado de la Tabla 19.

Se determina el error absoluto de cada prueba ejecutada mediante:

$$Error\ absoluto = Detecto\ Correctamente - Colocadas \quad (1)$$

La determinación del error relativo (ER) se lleva a cabo a través de la siguiente fórmula:

$$ER = \left| \frac{Valor\ Teorico - Valor\ Medido}{Valor\ Teorico} \right| \times 100 \quad (2)$$

Tabla 19. Eficacia del sistema.

	Colocadas	Detecto Correctamente	Error Absoluto	Error Relativo (ER)	Error Relativo Porcentual (%)
GRADO1	49	55	6	0,12	12,24
GRADO2	49	43	-6	0,12	12,24
GRADO 3	57	56	-1	0,02	1,75
GRADO 4	33	31	-2	0,06	6,06
GRADO 5	35	38	3	0,09	8,57
PROMEDIO					8,18

Con el error relativo porcentual calculado se obtiene la eficiencia que tiene el sistema para determinar cada grado de maduración de la pitahaya como resultado de este se tiene: grado 1: 87,76%, grado 2: 87,76%, grado 3: 98,25%, grado 4: 93,94%, y grado 5: 91,43%.

3.5 Confiabilidad del prototipo

La confiabilidad del sistema se establece al calcular el promedio ponderado de los errores porcentuales individuales.

$$\begin{aligned} \text{Confiabilidad del sistema} & \quad (3) \\ & = 100\% - \text{error promedio porcentual del sistema} \\ \text{Confiabilidad del sistema} & = 100\% - 8.18\% \\ \text{Confiabilidad del sistema} & = \mathbf{91.82\%} \end{aligned}$$

La confiabilidad del prototipo se fundamenta en la respuesta obtenida durante el entrenamiento y la precisión lograda por el algoritmo de visión artificial YOLO v5, utilizado para clasificar los grados de madurez de la pitahaya. La validación del algoritmo de visión artificial indica que el sistema presenta una tasa de precisión del 91.82%, asegurando así que el prototipo sea altamente confiable en la investigación realizada en el campo de la agricultura aplicando técnicas de visión artificial.

3.6 Presupuesto

Para el desarrollo del sistema inteligente de selección de cosecha en cultivo de pitahaya mediante visión artificial se emplearon diversos componentes, los cuales se detallan en la Tabla 20 que se muestra a continuación:

Tabla 20. Presupuesto.

Ítem	Detalle	Unidad	Cantidad	Valor Unitario	Valor Total
1	Cámara Web	1	1	30,00	30,00
2	Luces Led	1	2	4,00	8,00
3	Impresión Cabina	c/u	1	55,00	55,00
4	Material de oficina	c/u	1	20,00	20,00
5	Impresiones	c/u	200	0,05	10,00
6	Transporte	c/u	20	4,00	80,00
7	Alimentación	c/u	10	2,50	25,00
Subtotal					228,00
Imprevistos (5%)					14,40
Total					242,40

a. Precio de la mano de obra

La evaluación de los costos laborales se basa en el salario de un Ingeniero en Telecomunicaciones, de acuerdo al Ministerio de Trabajo, el salario anual de un Ingeniero Electrónico en Ecuador asciende a \$14,840, dividido por los 12 meses del año, este salario se estima en \$1236,66 mensuales, y puede ajustarse según el rango ocupado. Dado que el proyecto tiene una duración aproximada de 1 mes, el costo total de la mano de obra se estima en \$1236,66.

Considerando los valores calculados se obtiene el presupuesto total, sumando el presupuesto de hardware y el valor de la mano de obra.

$$\mathbf{Presupuesto\ total = Costo\ de\ materiales + mano\ de\ obra} \quad (4)$$

$$Presupuesto\ total = 242,40 + 1236,66$$

$$Presupuesto\ total = 1479.06$$

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La investigación ha llevado a cabo un análisis de los elementos de producción y las características determinantes de la calidad de las pitahayas. Este enfoque no solo ha abordado factores básicos como el crecimiento y la maduración, sino también ha explorado en profundidad elementos cruciales como el color de maduración de la fruta ya que así esta pigmentación permite la exportación y consumo nacional. La norma técnica NTE INEN 025 establece los grados de maduración de la pitahaya siendo cinco, de los cuales el grado tres y cuatro están aptos para exportación y el grado cinco para consumo nacional. Este análisis exhaustivo ha sentado las bases esenciales para el desarrollo y la implementación del sistema inteligente de selección de cosecha.
- En la investigación se ha identificado y evaluado de manera detallada las técnicas actuales de Visión Artificial aplicadas en cultivos. La elección de YOLOv5 para el entrenamiento del modelo ha sido crucial en la determinación de técnicas avanzadas de Visión Artificial aplicadas al cultivo de pitahaya, ya que este modelo incorpora mejoras en sus capas ocultas, esta implementación específica proporciona un conocimiento sólido sobre las capacidades y beneficios de YOLOv5 en el contexto agrícola, destacando su relevancia en el ámbito de la visión artificial.
- El sistema, basado en YOLOv5, evalúa la maduración de pitahayas en tres etapas: la etapa 1 incluye los grados 1 y 2 de maduración, la etapa 2 corresponde al grado 3, y la etapa 3 abarca los grados 4 y 5. El modelo de entrenamiento logró con éxito la clasificación, obteniendo una precisión del 99%. Para clasificar los grados 1 y 2, así como los grados 4 y 5 de maduración, se analizaron los porcentajes de color verde o amarillo mediante el historial global de cada imagen procesada. Este enfoque permite una evaluación precisa y detallada de la calidad de la cosecha, ofreciendo resultados cuantitativos sobre la distribución de los grados de maduración. Este proceso agiliza la toma

de decisiones en la selección de cosecha y facilita el seguimiento histórico de los resultados.

- La interfaz gráfica desarrollada en Python simplifica el acceso, la interpretación y la clasificación de los datos de la fruta, mejorando la usabilidad y la experiencia del usuario final. Proporciona una herramienta intuitiva para visualizar eficazmente los resultados del sistema de selección de cosecha. No solo facilita el acceso a los datos, sino que también presenta el número total de pitahayas visualizadas diariamente, que se van analizando de manera continua
- Se desarrolló un sistema en el que el usuario introduce seis pitahayas para su clasificación. La iluminación de la cámara de adquisición de imágenes con luz LED blanca resultó apropiada para la detección de los grados de maduración. Las pruebas de campo llevadas a cabo en la finca productora de pitahayas demostraron que el sistema posee una precisión total del 91.82% al clasificar la fruta según sus grados de maduración.

4.2 Recomendaciones

- Dada la importancia del color en la determinación de etapa de maduración de las pitahayas, se recomienda explorar técnicas avanzadas de análisis de color en futuras investigaciones. Esto podría incluir la aplicación de modelos de aprendizaje profundo específicos para la identificación precisa de matices y variaciones en el color de la fruta, mejorando aún más la capacidad del sistema para evaluar la calidad.
- Es necesario considerar la expansión del conjunto de datos para abordar variaciones estacionales y condiciones de iluminación que proporcione mayor cantidad de información del sistema.
- Se debe continuar monitoreando y ajustando el rendimiento del sistema a medida que se obtengan más datos de campo y retroalimentación del usuario. Este enfoque garantizará que el sistema se adapte a las dinámicas cambiantes del entorno agrícola, permitiendo ajustes o mejoras en tiempo real para mantener y mejorar su eficacia a lo largo del tiempo

- Se recomienda una exploración exhaustiva de opciones de optimización y escalabilidad para preparar el sistema para implementaciones a gran escala en entornos agrícolas extensos. Esto podría incluir consideraciones técnicas como la optimización de algoritmos, la gestión eficiente de recursos computacionales y la evaluación de tecnologías que permitan el manejo eficaz de grandes volúmenes de datos en tiempo real.
- Los resultados obtenidos podrían optimizarse con una base de datos más extensa que brinde información gráfica sobre los cinco grados de maduración de la pitaya amarilla, ya que en niveles de entrenamiento medios de visión artificial pueden surgir errores o confusiones en la detección.

REFERENCIAS BIBLIOGRÁFICAS

- [1] E. García y F. Fernando, «Agricultura de Presición,» Revista Ciencia y Tecnología, Argentina, 2008.
- [2] M. León y M. Gómez, «Evolución de la producción y comercio mundial de Frutas en el Mundo,» Cartagena, 2016.
- [3] La República, «Latinoamérica representa cerca de 13% de la producción agrícola a nivel mundial,» 29 07 2021. [En línea]. Available: <https://www.larepublica.co/globoeconomia/latinoamerica-representa-cerca-de-13-de-la-produccion-agricola-a-nivel-mundial-3208360#:~:text=En%20cuanto%20a%20temas%20agr%C3%ADcolas,explotaciones%20de%20la%20regi%C3%B3n%20se.>
- [4] Organización de las Naciones Unidas para la Alimentación y la Agricultura, «Ecuador en una mirada,» [En línea]. Available: [https://www.fao.org/ecuador/fao-en-ecuador/ecuador-en-una-mirada/es/.](https://www.fao.org/ecuador/fao-en-ecuador/ecuador-en-una-mirada/es/)

- [5] V. Andrade, «La tecnificación agrícola,» *El Telegrafo*, p. 1, 2017 Febrero 21.
- [6] J. B. Llecha, «Técnicas de control de calidad en fruta,» [En línea]. Available: <https://www.tdx.cat/bitstream/handle/10803/6877/CAPITOL2.pdf;jsessio>.
- [7] A. Carrera, «El Desarrollo de la Tecnología para la Industrialización de la Pitahaya (*Cereus triangularis* Haw) y su Incidencia en la Baja Oferta de Productos en el Mercado Local,» Ambato, 2011.
- [8] Organización de las Naciones Unidas para la Alimentación y la Agricultura, «Guía para prevenir y reducir la pérdida de frutas y hortalizas,» 2019. [En línea]. Available: <https://www.fao.org/family-farming/detail/es/c/1264841/>.
- [9] M. Delano y A. Collado, «CEPAL,» [En línea]. Available: <https://comunidades.cepal.org/ilpes/es/grupos/discusion/la-agricultura-del-futuro-cambios-y-desafios>.
- [10] J. Herrera, S. Medina, K. Beleño y O. Gualdrón, «Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial,» Colombia, 2016.
- [11] P. Neira, «Implementación de una estación prototipo con visión artificial, aplicado a la agricultura de precisión,» Cuenca, 2019.
- [12] V. Mazzia, A. Khaliq, F. Salvatti y M. Chiaberge, «Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application,» Turin, 2020.

- [13] N. Ismail y O. Maik, «Real-time visual inspection system for grading fruits using computer vision and deep learning techniques,» China, 2022.
- [14] W. Vivar y M. Vivas , «Diseño e Implementación de un Sistema de Control Automático con Visión Artificial y Redes Neuronales Destinado al Control de Calidad de Alimentos,» Guayaquil, 2022.
- [15] A. Garcés, «Sistema cuantificador de calidad de cultivo de manzana para monitoreo de la producción utilizando algoritmos de Aprendizaje Profundo con Visión Artificial y Segmentación de Instancias,» Ambato, 2022.
- [16] R. Bongiovanni, E. Chartuni, S. Best y Á. Roel, «AGRICULTURA DE PRECISIÓN: Integrando conocimientos para una agricultura moderna y sustentable.,» Procisur, Brasil, 2006.
- [17] Á. Llerena, «Agricultura digital en el cultivo de Pitahaya,» Telcombas S.A., Guayaquil, 2020.
- [18] Servicio de Acreditación Ecuatoriano, «Control de calidad para frutas y hortalizas,» 19 09 2018. [En línea]. Available: <https://www.acreditacion.gob.ec/control-calidad-para-frutas-y-hortalizas/>.
- [19] A. Quijije, «Estudio de Parámetros de Calidad y Característica Sensorial de Dos Variedades de Pitahaya Rosa (*Hylocereus undatus*), Pitahaya Amarilla (*Selenicereus megalanthus*) para su Aplicación en Procesos Agroindustriales,» Quevedo, 2021.
- [20] Norma Técnica Ecuatoriana NTE INEN 2003:2005, Frutas frescas, Pitahaya Amarilla. Requisitos, Quito: Instituto Ecuatoriano de Normalización, 2005.

- [21] Y. Vargas, J. Pico, A. Diaz, D. Sotomayor, A. Burbano, C. Caicedo, N. Parades, C. Congo, L. Tinoco, S. Bastidas, J. Chuquimarca, J. Macas y W. Viera, «Manual del Cultivo de Pitahaya,» La Joya de los Sachas, 2020.
- [22] P. Medina y F. Mendoza, «Elaboracion de Mermeladay Nectar a Partir de la Pulpa de Pitahaya y Determinacion de Capacidad Antioxidante por el Metodo DPPH (1,1 DIFENIL-2- PICRIL HIDRAZILA),» Guayaquil, 2017.
- [23] L. Rouhiainen, «Inteligencia Artificial,» alienta, España, 2018.
- [24] B. López, «Introducción a la Inteligencia Artificial,» Mexico, 2007.
- [25] P. Garcia, «Sistema de seguimiento de personas desde un dron,» España, 2022.
- [26] G. Zambrano, C. Parra, M. Marique y C. Bustacara, «Estacion de control de calidad por vision artificial para un centro de manufactura integrada por computador (CIM,» Pontificia Universidad Javeriana, 2009.
- [27] N. L. S. Palomino y U. Román, «Técnicas de Segmentación en Procesamiento Digital de Imágenes,» Lima.
- [28] B. Borella, «Introducción a la Visión Artificial. Procesos y Aplicaicones,» Madrid, 2022.
- [29] Intel, «What Is Machine Vision?,» 2021. [En línea]. Available: <https://www.intel.es/content/www/es/es/manufacturing/what-is-machine-vision.html>.

- [30] L. Medina , J. Mayorquín , J. Rodríguez y F. Morales, «Identificación de Componentes de Color con Visión Artificial y Redes Neuronales,» México, 2023.
- [31] A. Gonzalez, F. Martinez, A. Pernia, F. Alba, M. Castejon, J. Ordieros y E. Vergara, «Tecnicas y Algoritmos Basicos de de Vision Artificial,» Universidad de La Rioja, 2006.
- [32] R. Miranda, J. Solano y A. Méndez, «Introducción al Aprendizaje Automático con YOLO,» *Revista de la Facultad de Ingenierías y Tecnologías de la Información y Comunicación*, vol. 3, nº 2, 2019.
- [33] saturday.ai, «Posts tagged: YoloV5,» 09 marzo 2022. [En línea]. Available: <https://saturdays.ai/tag/yolov5/#:~:text=El%20modelo%20seleccionado%20fue%20el,para%20detectar%20objetos%20en%20im%C3%A1genes..>
- [34] Medium, «Medium: Breve reseña: YOLOv5 para la detección de objetos,» 26 06 2020. [En línea]. Available: <https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a>.
- [35] A. Laura, «Sistema Automático para el Control de la Calidad del Calzado Mediante Visión Artificial,» Ambato, 2023.
- [36] I. Challenger , Y. Díaz y R. Becerra, «El lenguaje de programación Pytho,» *Ciencias Holguín*, vol. XX, nº 2, pp. 1-13, 2014.
- [37] L. Larcher , P. Juárez, A. Ruggeri, E. BIASONI, C. Cattaneo y G. Villalba, «Ponderación de Calidad en Frutas Usando Técnicas de Visión Artificial para

la Estimación de Daños,» *Asociación Argentina de Mecánica Computacional*, vol. XXXII, pp. 2473-2484, 2013.

- [38] D. Figueroa y E. Roa , «Sistema de visión artificial para la identificación del estado de madurez de frutas (granadilla),» *Revista Redes de Ingeniería*, pp. 84-92, 2016.
- [39] L. Molina y C. Vargas, «Estudio e implementación de un sistema de control de calidad para la detección de contaminantes superficiales de diferentes tpos de frutas usando visión artificial,» Universidad de las Fuerzas Armadas, Latacunga, 2019.
- [40] P. Atencio, «Desarrollo de un sistema de visión artificial para el control de calidad del mango de azúcar de exportación,» Universidad del Magdalena , Santa Marta, 2009.
- [41] B. Nieto y J. Rangel, «Sistema de visión artificial para gestión de calidad del Banano Cavendish en etapa de postcosecha,» *Revista de Iniciación Científica*, vol. 8, nº 2, 2022.
- [42] E. Tapia, «Sistema de alerta del estado de maduración de alimentos frescos dentro de un refrigerador utilizando Inteligencia Artificial,» Universidad Autónoma de Querétaro Facultad de Ingeniería, Querétaro, 2023.
- [43] Primicias, «Cuatro empresas comienzan exportación de pitahaya a China,» 10 04 2023. [En línea]. Available: <https://www.primicias.ec/noticias/economia/pitahaya-exportaciones-china-ecuador/>.

- [44] Agrocalidad, «Control de calidad para frutas y hortalizas,» 03 02 2022. [En línea]. Available: <https://www.agrocalidad.gob.ec/exportaciones-de-pitahaya-crecieron-casi-60-en-2021/>.
- [45] Agrocalidad, «Pitahaya de Ecuador se exporta ya a 56 países: Perú se suma a la lista,» 15 11 2022. [En línea]. Available: <https://www.agrocalidad.gob.ec/pitahaya-de-ecuador-se-exporta-ya-a-56-paises-peru-se-suma-a-la-lista/>.
- [46] S. Amaya, D. Pulgarín y I. Torres, «Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM),» *Lámpsakus*, nº 15, pp. 43-50, 2016.
- [47] Trust, «Cámara web full HD de 1080P GXT1160 VERO,» 2023. [En línea]. Available: <https://www.trust.com/es/product/22397-gxt-1160-vero-streaming-webcam#>.
- [48] Pinsfot, «Cod. 045 WebCam con Micrófono D-Bugg M26 / 480p / USB2.0 + Jack 3.5mm/ Modo Night Visión,» 2023. [En línea]. Available: <https://pinsoft.ec/cod-045-webcam-con-micrfono-d-bugg-m26-480p-usb2-0-jack-3-5mm/p-6154.htm>.
- [49] R. Pi, «Raspberry Pi Camera Module 2 NoIR,» 2023. [En línea]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html..>
- [50] S. Hensel, «B&R a member of the ABB Group,» 2023. [En línea]. Available: <https://www.br-automation.com/es/sobre-nosotros/sala-de-prensa/technology-highlights/hagase-la-luz/>.

- [51] Philips, Lámparas LED Essential, Signify Holding, 2023.
- [52] Philips, T5 Essential Tubo lineal fluorescente, Perú, 2018.
- [53] Lámparadirecta.es, Especificaciones Philips Incandescente Estándar 25W E27 230V P45, España, 2023.
- [54] D. Robayo, «Calidad en el desarrollo de software desde una perspectiva bibliométrica en el software desde una perspectiva bibliométrica en el,» *INNOVA Research Journal*, vol. 8, n° 3, pp. 173-188, 2023.
- [55] AVENUE Learning, «¿Qué es el lenguaje de programación C++?,» julio 2012. [En línea]. Available: <https://www.avenuglobal.com/noticias/estudiar-desarrollo-de-software-que-es-el-lenguaje-de-programacion-c..>
- [56] A. Ginger, Estudio comparativo de lenguajes multiplataforma JAVA, PHP, que permitan garantizar un óptimo desempeño al desarrollar aplicaciones de escritorio, Babahoyo: Universidad Técnica de Babahoyo, 2022.
- [57] J. Cuaresma, «mycompiler, para probar código en varios lenguajes directamente desde la web,» 2019. [En línea]. Available: <https://unividaful.edu.co/bienestar/mycompiler-para-probar-codigo-en-varios-lenguajes-directamente-desde-la-web/#:~:text=Hoy%20hablaremos%20de%20una%20nueva,elegir%20qu%C3%A9%20lenguaje%20queremos%20probar..>
- [58] G. Baume, Breve introducción a Google Colab, 2021.

- [59] K. Kim , K. Kim y S. Jeong, «Application of YOLO v5 and v8 for Recognition of Safety Risk Factors at Construction Sites,» *Sustainability*, vol. 15, n° 15179, pp. 1-17, 2023.
- [60] G. Chandan, A. Jain , H. Jain y Mohana, «Real Time Object Detection and Tracking Using Deep Learning and OpenCV,» India, Coimbatore, 2020, pp. 1305-1308.
- [61] TensorFlow, «Guía TensorFlow Lite,» 2023. [En línea]. Available: <https://www.tensorflow.org/lite/guide/inference?hl=es-419..>
- [62] P. López, Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL, Texcoco: Universidad Autónoma del Estado de México, 2016.
- [63] Microsoft, Microsoft SQL Server 2017, 2017.
- [64] O. Olorunshola, Irhebhude y Ewwiekpaefe, «A comparative study of YOLOv5 and YOLOv7 Object Detection Algorithms,» *Journal of Computing and Social Informatics*, vol. 2, n° 1, pp. 1-12, 2023.
- [65] Organización de las Naciones Unidas para la Alimentación y la Agricultura, «Fichas técnicas Productos frescos de frutas.,» [En línea]. Available: <http://www.fao.org/3/a-au173s.pdf..>

ANEXOS

Anexo A. Entrevista realizada a jefe y empleados de la finca

¿Cuál es el color de pigmentación de la piel de la pitahaya para Estados Unidos, Perú y consumo nacional?

¿Qué calibre es recomendado por el INEN para exportar la pitahaya?

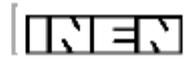
¿Conoce usted la norma INEN para exportación?

¿Qué tamaño es el más consumido por población extranjera?

¿Qué particularidad presenta la pitahaya para ser separada y no ser exportada?

¿En qué tiempo de productividad la pitahaya esta apta para la exportación y expendio en el mercado nacional?

Anexo B. Norma Técnica Ecuatoriana NTE INEN 2003:2005



INSTITUTO ECUATORIANO DE NORMALIZACIÓN

Quito - Ecuador

NORMA TÉCNICA ECUATORIANA

NTE INEN 2 003:2005

FRUTAS FRESCAS. PITAJAYA AMARILLA. REQUISITOS.

Primera Edición

FRESH FRUITS. PITAHAYA. SPECIFICATIONS.

First Edition

Anexo C. Datasheet Cámara Web Trust Gaming GXT 1160 Vero.

Trust.



Trust Gaming GXT 1160 Vero Cámara web Full HD de 1080 p - Negro

Elegante cámara web Full HD de 1080 p con micrófono incorporado

Article number: 22397
Barcode: 8713439223972
GTIN: 08713439223972

Core content

Brand	Trust Gaming
Product type	Cámara web Full HD de 1080 p
Colour	Negro
Key features	- Imágenes nítidas con resolución HD de 1920 x 1080 a 30 fotogramas por segundo - Conveniente micrófono incorporado, ideal para chats de vídeo - Perfecta para Skype, YouTube y streaming en línea. - Práctico pedestal dotado de pinza integrada, ideal para pantallas de ordenadores portátiles y superficies planas - Balance de blancos automático: vídeo nítido en cualquier condición de luz - Enfoque fijo para una imagen clara y nítida
Package contents	- Cámara web Full HD
System requirements	- Puerto USB - Windows 10, 11

Anexo D. Led Panel RD.

SYLVANIA

LED Panel

LED PANEL RD 3W WW 100-240V
P27359



Luminaria tipo Panel LED con driver independiente. Montaje de incrustar en cielo raso, con clip de fijación. Proyección uniforme de la luz, reduce los costos de consumo de energía y de mantenimiento.

CARACTERÍSTICAS

Diseño moderno con fuente LED SMD y difusor opalizado
Ultra delgado y liviano con disipador de calor integrado
Ganchos de sujeción con resorte para fácil instalación

APLICACIONES

Adecuado para aplicaciones con altura limitada de instalación entre cielo raso y placa
Iluminación general residencial
Zonas comunes

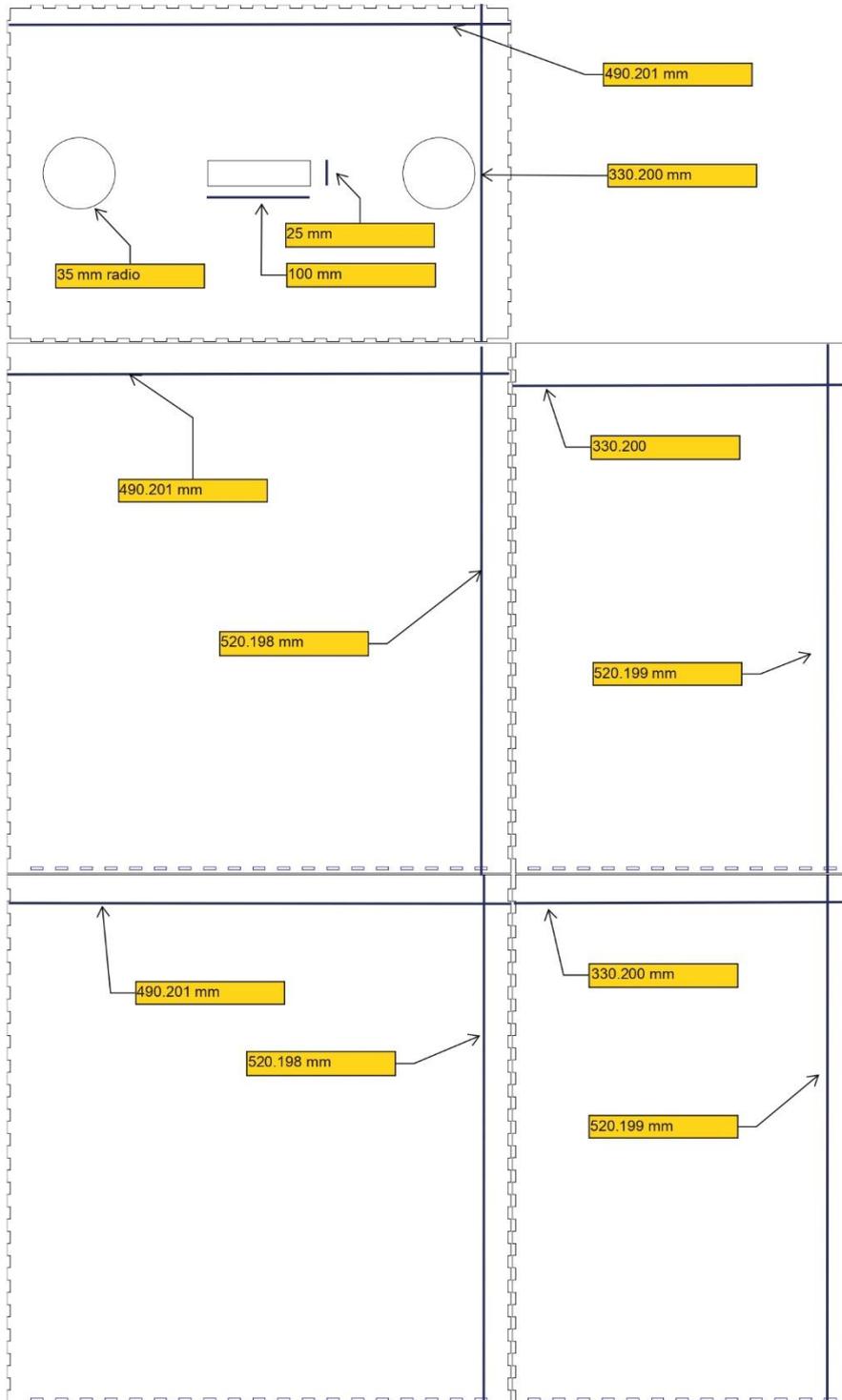


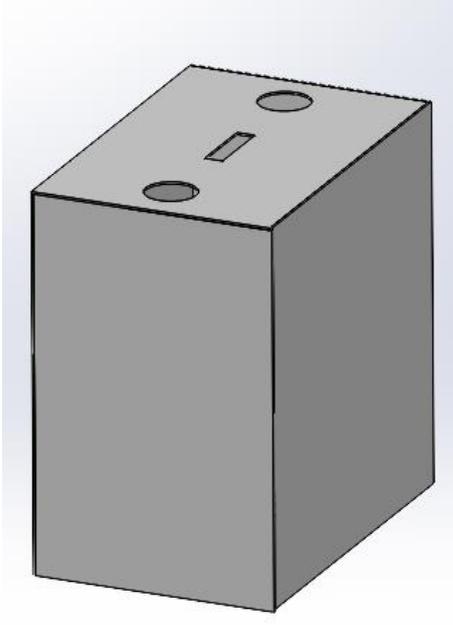
Ultra
Delgado

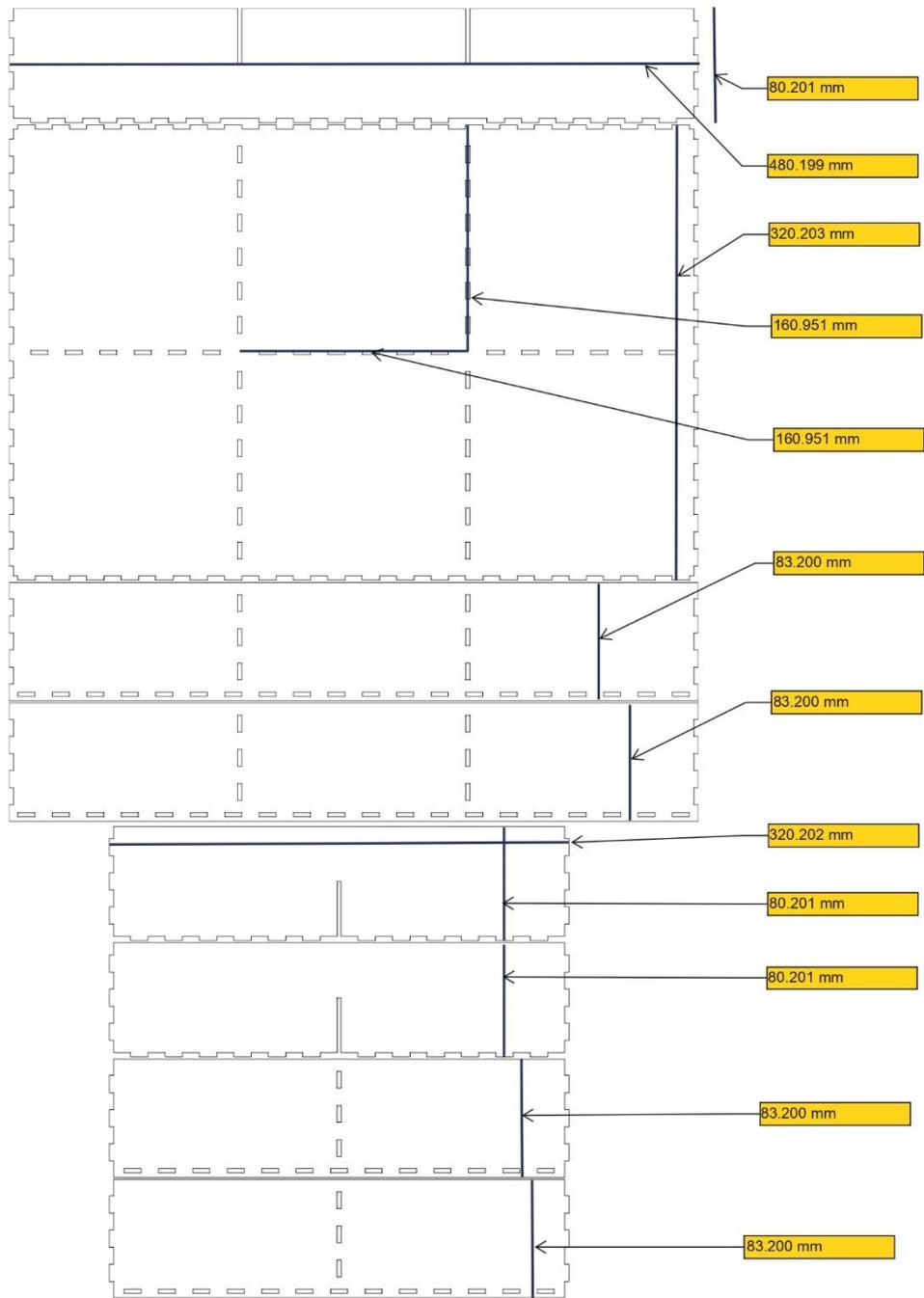


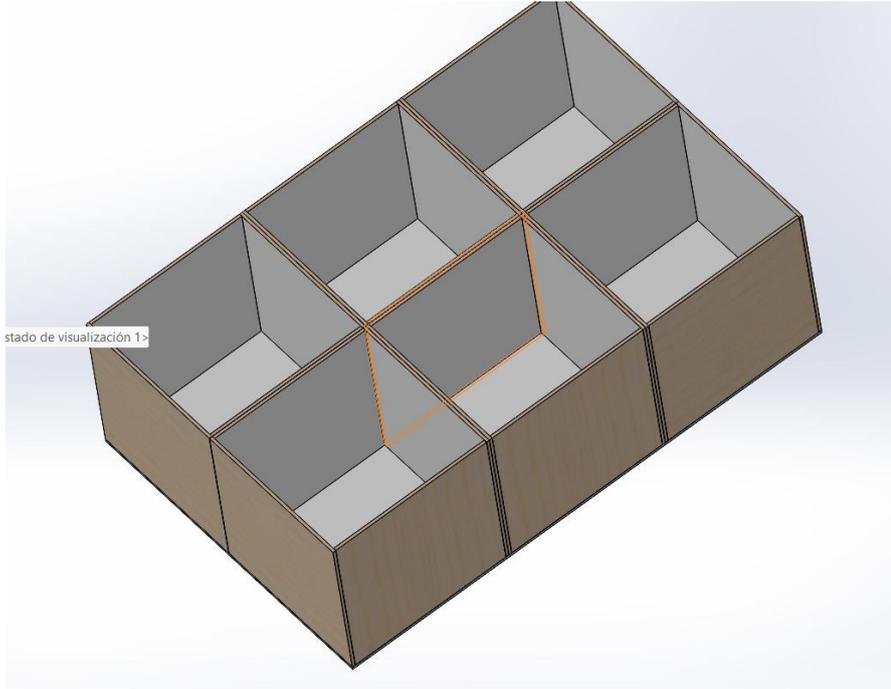
DATOS ÓPTICOS		DATOS FÍSICOS		DATOS ELÉCTRICOS	
Temperatura de color	3000 K (WW)	Acabado	Bianco	Potencia de entrada	3 W
Flujo luminoso	110 lm	Grado de protección IP	IP20	Tensión de operación	100-240 V 50/60 Hz
Ángulo de apertura	120°	Dimensiones (DxH)	Ø83x20 mm	Corriente de entrada	0.025 A @ 120 V
Tipo de distribución	Directa simétrica	Tipo de montaje	Incrustar	Factor de potencia	>0.4
Reproducción de color (IRC)	70	Chasis	Aluminio + PC	Distorsión armónica (THD)	
Vida útil	25000 h L70	Óptica	Difusor opalizado	Tipo de driver	Independiente CC
Eficacia	37 lm/W	Temperatura de operación Ta	-20°C ~ +50°C	Atenuable	NO

Anexo E. Diseño de la Cabina de Visión Artificial.









Anexo F. Código de Python.

```
#Importación de librerías
import tkinter #libreria de kit de herramientas de GUI Tk
import cv2 #libreria para trabajo con imagenes
import time # módulo que proporciona varias funciones relacionadas con
el tiempo
import torch #cálculos numéricos mediante programación de tensores para
aprendizaje profundo
import numpy as np # cálculo numérico y el análisis de datos,
especialmente para un gran volumen de datos
import tkinter.font as font #proporciona la clase Font para crear y
usar fuentes con nombre
from tkinter import ttk #el módulo ttk proporciona acceso al conjunto
de widgets temático
from tkinter import *
import json as json #formato de intercambio de datos ligero
from PIL import Image, ImageTk
from urllib.request import Request, urlopen
import requests #peticiones para metodos
from tkinter import messagebox #Creacion de una ventana de mensaje con
un mensaje especificado por la aplicación

#declaración de variables
nueva_imgCam1 = None
historial_etapa1 = 0
historial_etapa2 = 0
historial_etapa3 = 0
historial_etapa4 = 0
historial_etapa5 = 0

#Conexión a YOLO por ubicacion del archivo
try:

    time.sleep(1)
    model = torch.hub.load('ultralytics/yolov5', 'custom',
                           path='C:/Users/USER/Documents/Tesis/sistema/Pitah
aya1/best.pt')
except:
    print("Error")

#Ventana principal de interfaz
ventana = tkinter.Tk()
ventana.title("Lenin Gutiérrez - Sistema de control de calidad de
pitahaya")
ventana.config(bg="grey")
ventana.geometry("800x600+100+100")
```

```

ventana.iconbitmap('icono.ico')
datos = tkinter.Toplevel(ventana)
datos.title("Lenin Gutiérrez - Sistema de control de calidad de
pitahaya")
datos.config(bg="#d9d9d9")
datos.iconbitmap('icono.ico')

#minimizar ventanas cuando se abre ventanas secundarias
def minimizar_ventanas_secundarias():
    datos.iconify()

minimizar_ventanas_secundarias()

#Enviar datos a la base por medio de URL
def enviar(dir):
    request = Request(dir)
    json = urlopen(request).read().decode()
    return

#Control de usuario (usuario:Lenin y contraseña: uta)
def controlIngreso():
    texto_usuario = entryInicioUsuario.get()
    texto_contrasena = entryInicioContrasena.get()
    if texto_usuario=="Lenin" and texto_contrasena=="uta":
        ventanaPrincipal()
    else:
        messagebox.showerror("Ingreso Incorrecto", "Datos incorrectos")

#Ventana de la caratula
def abrirCaratula():
    ventana.state(newstate="zoomed")
    datos.state(newstate="withdraw")

#Ventana secundaria del sistema
def ventanaPrincipal():
    datos.state(newstate="zoomed")
    ventana.state(newstate="withdraw")

#Petición por una URL para traer los datos del servidor
def consultaHistorial():
    request =
Request("http://localhost:8080/pitahaya/verAcumulado.php")
    json = urlopen(request).read().decode()
    return json

#Inicializacion de variables globales en formato JSON
def enviarHistorial():

```

```

global historial_etapa1
global historial_etapa2
global historial_etapa3
global historial_etapa4
global historial_etapa5
global etapa1_contador
global etapa2_contador
global etapa3_contador
global etapa4_contador
global etapa5_contador
valores_a_enviar = {
    "etapa1": etapa1_contador,
    "etapa2": etapa2_contador,
    "etapa3": etapa3_contador,
    "etapa4": etapa4_contador,
    "etapa5": etapa5_contador
}
datos_json = json.dumps(valores_a_enviar)
try:
    response =
requests.post("http://localhost:8080/pitahaya/enviar.php",
data=datos_json)
    print("Respuesta desde PHP:", response.text)
    if response.status_code == 200:
        print("Datos enviados correctamente al servidor.")
    else:
        print(f"Error al enviar datos al servidor. Código de
estado: {response.status_code}")
    except Exception as e:
        print(f"Error al enviar datos al servidor: {str(e)}")

#Funcion que permite mostrar el historial con URL establecida en enviar
historial
def mostrarHistorial():
    global historialGrado1
    global historialGrado2
    global historialGrado3
    global historialGrado4
    global historialGrado5
    global historial_etapa1
    global historial_etapa2
    global historial_etapa3
    global historial_etapa4
    global historial_etapa5

    tDatos = consultaHistorial()
    historial = json.loads(tDatos)

```

```

if historial and len(historial) > 0:
    ultimo_registro = historial[-1]
    historial_etapa1 = ultimo_registro.get('etapa1', 0)
    historial_etapa2 = ultimo_registro.get('etapa2', 0)
    historial_etapa3 = ultimo_registro.get('etapa3', 0)
    historial_etapa4 = ultimo_registro.get('etapa4', 0)
    historial_etapa5 = ultimo_registro.get('etapa5', 0)
    historialGrado1.config(text=f"Grado 1: {historial_etapa1}")
    historialGrado2.config(text=f"Grado 2: {historial_etapa2}")
    historialGrado3.config(text=f"Grado 3: {historial_etapa3}")
    historialGrado4.config(text=f"Grado 4: {historial_etapa4}")
    historialGrado5.config(text=f"Grado 5: {historial_etapa5}")
    print("hay dat")
    print(historial_etapa1)
else:
    historialGrado1.config(text="Grado 1: 0")
    historialGrado2.config(text="Grado 2: 0")
    historialGrado3.config(text="Grado 3: 0")
    historialGrado4.config(text="Grado 4: 0")
    historialGrado5.config(text="Grado 5: 0")
    print("no hay dat")
    print("0")

#conexion con la camara web y enviar al modelo
def capturar():
    global lblImgCam1
    global nueva_imgCam1
    global deteccionGrado1
    global deteccionGrado2
    global deteccionGrado3
    global deteccionGrado4
    global deteccionGrado5
    global historial_etapa1
    global historial_etapa2
    global historial_etapa3
    global historial_etapa4
    global historial_etapa5
    global historialGrado1
    global historialGrado2
    global historialGrado3
    global historialGrado4
    global historialGrado5
    global etapa1_contador
    global etapa2_contador
    global etapa3_contador
    global etapa4_contador
    global etapa5_contador
    cap = cv2.VideoCapture(0)

```

```

mostrarHistorial()

leido0, frame0 = cap.read()
cap.release()
cv2.imwrite("camara.jpg", frame0)
image_path = "camara.jpg"
#frame1 = cv2.imread(image_path)
im1 = Image.open(image_path)
model.conf = 0.80
print("paso1")
#detect = model(frame0, size=640)
detect = model(im1, size=640)
info = detect.pandas().xyxy[0]
print("paso2")
arr = np.asarray(info.confidence.array)
detectar = len(arr)
etapa1_contador = 0
etapa2_contador = 0
etapa3_contador = 0
etapa4_contador = 0
etapa5_contador = 0
print("paso3")
print("arr")
#tratamiento de imagen para color verde Etapa1, Etapa2, Etapa3 ,
Etapa4 y Etapa 5
for i in range(len(info)):
    if len(info.iloc[i]) >= 6:
        clase = int(info.iloc[i]['class'])

        if clase == 0:
            print(clase)
            objeto = im1.crop((info.iloc[i]['xmin'],
info.iloc[i]['ymin'], info.iloc[i]['xmax'], info.iloc[i]['ymax']))
            porcentaje_verde = tratamiento_etapa1(objeto)
            if porcentaje_verde>30: #parametro de NTE INEN
2003:2005 porcentaje de color verde
                etapa2_contador += 1
            else:
                etapa1_contador += 1
        elif clase == 1:
            print(clase)
            etapa3_contador += 1
        elif clase == 2:
            print(clase)
            objeto = im1.crop((info.iloc[i]['xmin'],
info.iloc[i]['ymin'], info.iloc[i]['xmax'], info.iloc[i]['ymax']))

```

```

        porcentaje_amarillo, porcentaje_verde =
tratamiento_etapa3(objeto)
        print(porcentaje_amarillo)
        if porcentaje_amarillo>44: #parametro de NTE INEN
2003:2005 porcentaje de color amarillo
            etapa5_contador += 1
        else:
            etapa4_contador += 1

deteccionGrado1.config(text=f"Grado 1: {etapa1_contador}")
deteccionGrado2.config(text=f"Grado 2: {etapa2_contador}")
deteccionGrado3.config(text=f"Grado 3: {etapa3_contador}")
deteccionGrado4.config(text=f"Grado 4: {etapa4_contador}")
deteccionGrado5.config(text=f"Grado 5: {etapa5_contador}")
if detectar > 0:
    arr = info.to_numpy()
    rendered_image = np.squeeze(detect.render())
    rendered_image_rgb = cv2.cvtColor(rendered_image,
cv2.COLOR_BGR2RGB)
    cv2.imwrite("resultado.png", rendered_image_rgb)
    nueva_imagen = Image.open('resultado.png')
    nueva_imagen = nueva_imagen.resize((1200, 400))
    nueva_imgCam1 = ImageTk.PhotoImage(nueva_imagen)
    lblImgCam1.configure(image=nueva_imgCam1)
else:
    try:
        nueva_imagen = Image.open('pitahayaPrueba2.jpg')
        nueva_imagen = nueva_imagen.resize((1200, 400))
        nueva_imgCam1 = ImageTk.PhotoImage(nueva_imagen)
        lblImgCam1.configure(image=nueva_imgCam1)
        lblImgCam1.image = nueva_imgCam1
    except Exception as e:
        print("Error al procesar la imagen:", str(e))
historial_etapa1 = int(historial_etapa1) + etapa1_contador
historial_etapa2 = int(historial_etapa2) + etapa2_contador
historial_etapa3 = int(historial_etapa3) + etapa3_contador
historial_etapa4 = int(historial_etapa4) + etapa4_contador
historial_etapa5 = int(historial_etapa5) + etapa5_contador
historialGrado1.config(text=f"Grado 1: {historial_etapa1}")
historialGrado2.config(text=f"Grado 2: {historial_etapa2}")
historialGrado3.config(text=f"Grado 3: {historial_etapa3}")
historialGrado4.config(text=f"Grado 4: {historial_etapa4}")
historialGrado5.config(text=f"Grado 5: {historial_etapa5}")

enviarHistorial()

#Control de Etapa1
def tratamiento_etapa1(objeto):

```

```

img_cv2 = np.array(objeto)
img_cv2 = cv2.cvtColor(img_cv2, cv2.COLOR_RGB2BGR)
img_hsv = cv2.cvtColor(img_cv2, cv2.COLOR_BGR2HSV)
lower_green = np.array([40, 40, 40]) #parametro de color verde RGB
upper_green = np.array([80, 255, 255]) #parametro de color verde en
escala de grises
mask_green = cv2.inRange(img_hsv, lower_green, upper_green)
area_total = objeto.size[0] * objeto.size[1]
area_verde = cv2.countNonZero(mask_green)
porcentaje_verde = (area_verde / area_total) * 100 #procesamiento
de imagen mediante area verde
print(f"Porcentaje de Verde: {porcentaje_verde:.2f}%")

return porcentaje_verde

#Control Etapa2
def tratamiento_etapa3(objeto):
    img_cv2 = np.array(objeto)
    img_cv2 = cv2.cvtColor(img_cv2, cv2.COLOR_RGB2BGR)
    img_hsv = cv2.cvtColor(img_cv2, cv2.COLOR_BGR2HSV)
    lower_yellow = np.array([20, 100, 100]) #parametro de color
    amarillo RGB
    upper_yellow = np.array([30, 255, 255]) #parametro de color
    amarillo en escala de grises
    lower_green = np.array([40, 40, 40]) #parametro de color verde RGB
    upper_green = np.array([80, 255, 255]) #parametro de color verde en
    escala de grises
    lower_brown = np.array([10, 60, 20]) #parametro de color cafe RGB
    upper_brown = np.array([20, 255, 255]) #parametro de color cafe en
    escala de grises
    mask_yellow = cv2.inRange(img_hsv, lower_yellow, upper_yellow)
    mask_green = cv2.inRange(img_hsv, lower_green, upper_green)
    mask_brown = cv2.inRange(img_hsv, lower_brown, upper_brown)
    area_total = objeto.size[0] * objeto.size[1]
    area_amarillo = cv2.countNonZero(mask_yellow)
    area_verde = cv2.countNonZero(mask_green)
    area_cafe = cv2.countNonZero(mask_brown)
    porcentaje_amarillo = (area_amarillo / area_total) * 100
#procesamiento de imagen mediante area amarillo
porcentaje_verde = (area_verde / area_total) * 100 #procesamiento
de imagen mediante area verde
porcentaje_cafe = (area_cafe / area_total) * 100 #procesamiento de
imagen mediante area cafe
print(f"Porcentaje de Amarillo: {porcentaje_amarillo:.2f}%")
print(f"Porcentaje de Verde: {porcentaje_verde:.2f}%")
print(f"Porcentaje de Café: {porcentaje_cafe:.2f}%")
img_cafe = cv2.bitwise_and(img_cv2, img_cv2, mask=mask_brown)

```

```

    img_cafe_pil = Image.fromarray(cv2.cvtColor(img_cafe,
cv2.COLOR_BGR2RGB))
    img_cafe_pil.save("proceso3.png")

    return porcentaje_amarillo, porcentaje_verde

#deteccion actual de las sistema de las pitahayas
labelTituloDeteccion = tkinter.Label(
    datos, text="Detección Actual", font=('Times 18'))
labelTituloDeteccion.place(x=0, y=430, relwidth=0.5, height=40)

#historial acumulado de las sistema de las pitahayas
labelTtituloHistorial = tkinter.Label(
    datos, text="Historial", font=('Times 18'))
labelTtituloHistorial.place(relx=0.5, y=430, relwidth=0.5, height=40)

deteccionGrado1 = tkinter.Label(
    #datos, text="Grado 1: 0", font=('Times 15'), anchor='w',
justify='left')
    datos, text="Grado 1: 0", font=('Times 15'))
deteccionGrado1.place(x=0, y=460, relwidth=0.5, height=40) #parametro
para deteccion de Etapa1 mediante area

historialGrado1 = tkinter.Label(
    datos, text="Grado 1: 0", font=('Times 15'))
historialGrado1.place(relx=0.5, y=460, relwidth=0.5, height=40)
#parametro para deteccion de Etapa1 mediante area

deteccionGrado2 = tkinter.Label(
    datos, text="Grado 2: 0", font=('Times 15'))
deteccionGrado2.place(x=0, y=490, relwidth=0.5, height=40) #parametro
para deteccion de Etapa2 mediante area

historialGrado2 = tkinter.Label(
    datos, text="Grado 2: 0", font=('Times 15'))
historialGrado2.place(relx=0.5, y=490, relwidth=0.5, height=40)
#parametro para deteccion de Etapa2 mediante area

deteccionGrado3 = tkinter.Label(
    datos, text="Grado 3: 0", font=('Times 15'))
deteccionGrado3.place(x=0, y=520, relwidth=0.5, height=40) #parametro
para deteccion de Etapa3 mediante area

historialGrado3 = tkinter.Label(
    datos, text="Grado 3: 0", font=('Times 15'))
historialGrado3.place(relx=0.5, y=520, relwidth=0.5, height=40)
#parametro para deteccion de Etapa3 mediante area

```

```

deteccionGrado4 = tkinter.Label(
    datos, text="Grado 4: 0", font=('Times 15'))
deteccionGrado4.place(x=0, y=550, relwidth=0.5, height=40) #parametro
para deteccion de Etapa4 mediante area

historialGrado4 = tkinter.Label(
    datos, text="Grado 4: 0", font=('Times 15'))
historialGrado4.place(relx=0.5, y=550, relwidth=0.5, height=40)
#parametro para deteccion de Etapa4 mediante area

deteccionGrado5 = tkinter.Label(
    datos, text="Grado 5: 0", font=('Times 15'))
deteccionGrado5.place(x=0, y=580, relwidth=0.5, height=40) #parametro
para deteccion de Etapa5 mediante area

historialGrado5 = tkinter.Label(
    datos, text="Grado 5: 0", font=('Times 15'))
historialGrado5.place(relx=0.5, y=580, relwidth=0.5, height=40)
#parametro para deteccion de Etapa5 mediante area

#imagen de pitahayas que aparecera anten de capturar la imagen por
camara web
image1 = Image.open('pitahayaPrueba2.JPG')
img1 = image1.resize((1200, 500))
imgCam1 = ImageTk.PhotoImage(img1)
lblImgCam1 = tkinter.Label(datos, image=imgCam1)
lblImgCam1.place(x=0, y=0, relwidth=1, relheight=0.6)

subVen = tkinter.LabelFrame(ventana, text='')
subVen.place(relx=0, rely=0, relwidth=1, relheight=1)

# imagen de datos de la caratula
image_path = 'caratulaIniciarSesion.png'
image = Image.open(image_path)
width, height = ventana.winfo_screenwidth(),
ventana.winfo_screenheight()
resized_image = image.resize((800, 600))
tk_image = ImageTk.PhotoImage(resized_image)

imgLogo = tkinter.Label(subVen, image=tk_image, bg='#000')
imgLogo.place(x=0, y=0, relwidth=1, relheight=1)

entryInicioUsuario = tkinter.Entry(ventana, font=('Arial', 16))
entryInicioUsuario.place(relx=0.7, rely=0.42, anchor='center',
relwidth=0.4, height=40)

entryInicioContrasena = tkinter.Entry(ventana, font=('Arial', 16),
show='*')

```

```
entryInicioContrasena.place(relx=0.7, rely=0.6, anchor='center',
relwidth=0.4, height=40)

#boton de ingreso al sistema
botonIngresar = tkinter.Button(ventana, text="INGRESAR",
command=controlIngreso, fg="#000000", bg="#ffffff")
botonIngresar.place(relx=0.5, rely=0.85, anchor='s', relwidth=0.2,
height=40)
botonIngresar['font'] = font.Font(size=18, weight="bold")

#boton de capturar la imagen para proceso
botonCapturar = tkinter.Button(datos, text="CAPTURAR",
command=capturar, fg="#000000", bg="#ffffff")
botonCapturar.place(relx=0.5, rely=1.0, anchor='s', relwidth=0.2,
height=40)
botonCapturar['font'] = font.Font(size=18, weight="bold")

ventana.mainloop()
```

Anexo G. Posición de la cámara web en el sistema de visión artificial.

