



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR,
MODALIDAD DE PROYECTO DE INVESTIGACIÓN**

TEMA:

“PROGRAMA COMPUTACIONAL PARA CLASIFICAR AGUACATES”

Trabajo de titulación previo a la obtención del título de: *Ingeniero en Mecatrónica*

Línea de investigación: *Prototipos Industriales*

Autor: *Gómez Astudillo Jefferson David*

Director: *PhD. Ojeda Peña David Alberto*

Asesor: *PhD. García Santilla Iván Danilo*

Ibarra – 2023



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CEDULA DE IDENTIDAD:	1003985114		
APELLIDOS Y NOMBRES:	Gómez Astudillo Jefferson David		
DIRECCIÓN:	Av. Atahualpa y Cap. José Espinoza de los Monteros 9-57		
EMAIL:	jdgomeza1@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0997009035

DATOS DE LA OBRA	
TÍTULO:	“PROGRAMA COMPUTACIONAL PARA CLASIFICAR AGUCATES”
AUTOR:	Gómez Astudillo Jefferson David
FECHA:	13 /07 /2022
SOLO PARA TRABAJOS DE TITULACIÓN	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
DIRECTOR:	PhD. David Ojeda

CONSTANCIAS

El autor manifiesta que la obra, objeto de la presente autorización, es original y se la desarrollo sin violar derechos de autoridades de terceros, por lo tanto, la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 13 días del mes de julio de 2023.

EI AUTOR

Firma:



Nombre: Gómez Astudillo Jefferson David

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 13 de julio de 2023

PhD. David Ojeda

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de titulación, el mismo que se ajusta a las normas vigentes de la Unidad Académica de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



.....
PhD. David Ojeda

C.C.: 1757898489

APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación "PROGRAMA COMPUTACIONAL PARA CLASIFICAR AGUCATES" elaborado por Gómez Astudillo Jefferson David, previo a la obtención del título del INGENIERO EN MECATRÓNICA, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



.....
PhD. David Ojeda

C.C: 1757898489



.....
PhD. Iván García

C.C.: 1002292603

DEDICATORIA

A mis padres, Ángel y Elsa, que gracias a su amor infinito lograron plasmar en mi todas sus enseñanzas, por brindarme su apoyo incondicional a pesar de las adversidades y de los errores que pude cometer, por siempre estar a mi lado y no dudar ni un segundo en ofrecer su apoyo y palabras de aliento, haciéndome ver que la vida es dura, pero muy justa con las personas que hacen las cosas con humildad y amor verdadero, no podría haber pedido mejores padres que ustedes.

A mis hermanos Ángel, Edgar, Edison, Mery, Sofí, Xavi, por confiar en mi y en mis capacidades por compartir toda la vida llena de buenos y malos momentos, pero siempre juntos como nuestros padres nos lo enseñaron, estén seguros de que siempre podrán contar con su hermano incondicionalmente.

A mi hija Evolet Isabel motor de mi vida, alegría de mi vida y luz de mis ojos, que ya hace 10 años sin darse cuenta le dio un motivo y razón a mi existencia, todo mi esfuerzo es por usted mi amor, espero poder plasmar en usted que todo lo que se proponga con trabajo y dedicación se puede lograr. Le amo infinitamente mi vida.

A mis sobrinos Jean Carlos, Karen, Sebastián, Ángel, Edgar, Emir, Jadiel, Israel, que gracias a sus ocurrencias y alegrías le dan una razón más a mi vida para seguir luchando, gracias por hacerme recordar lo hermosa que es la niñez, sepan que siempre van a contar con su tío.

A Gaby que ha estado en los momentos donde decaía y supo apoyarme y levantarme con paciencia y cariño.

De corazón.

David

AGRADECIMIENTO

A mi tutor PhD. David Ojeda sin usted y sus virtudes, su paciencia y constancia este trabajo no lo hubiese logrado tan fácil. Sus consejos fueron siempre útiles cuando no salían de mi pensamiento las ideas para escribir lo que hoy he logrado. Usted formó parte importante de esta historia con sus aportes profesionales que lo caracterizan.

A mis docentes que gracias a su preparación y experiencia me supieron guiar de la mejor manera en el transcurso de estos años, Gracias por brindarme sus conocimientos de manera profesional e invaluable.

A la Universidad Técnica del Norte por permitirme ser parte de esta noble institución siempre formando entes profesionales y con valores.

A mis compañeros y amigos de viaje por compartir tantos momentos y no puedo dejar de recordar cuantas tardes y horas de trabajo nos juntamos a lo largo de nuestra formación siempre estaré agradecido por su apoyo y constancia, al estar en las horas más difíciles, por compartir horas de estudio.

A mis amigos Daniela G, Alison H, Damián V, Edwin E, Cristian R por siempre confiar en mí, por su lealtad y amistad incondicional y sincera, siempre les estaré agradecido.

A nuestro señor Jesucristo el cual me permitió refugiarme en su manto en mis momentos de decadencia, llenándome de amor y sabiduría para tomar mis decisiones.

David

RESUMEN EJECUTIVO

El presente trabajo de investigación presenta el desarrollo de un algoritmo basado en visión artificial de código abierto que permite clasificar y estimar el peso de los aguacates Hass en tiempo real a través de videos. El enfoque principal del algoritmo se basa en analizar las dimensiones aproximadas de longitud y anchura del fruto. El proceso comienza capturando el video en tiempo real y luego se aplican técnicas de procesamiento de imágenes para detectar colores, generar contornos, analizar dimensiones y, en base a estos resultados, clasificar el aguacate.

Para garantizar la precisión del algoritmo, se llevaron a cabo diversas pruebas en un entorno controlado para evitar la interferencia de factores externos, como la luz que podría alterar los colores captados por la cámara. Durante las pruebas, se utilizaron 30 frutas y se compararon las mediciones del algoritmo con los resultados de una balanza digital, siguiendo los criterios establecidos por la normativa ecuatoriana NTE INEN 1755, 2ª Edición. Los resultados mostraron un error porcentual promedio de estimación de peso del 0,06% y una eficiencia del algoritmo del 96,97% en las pruebas de clasificación de aguacates Hass.

Palabras clave: Visión Artificial, Tratamiento de imágenes, Python, Clasificación del aguacates, Aguacates tipo Hass, Estimación de Peso.

ABSTRACT

This research paper presents the development of an algorithm based on open source artificial vision that allows classifying and estimating the weight of Hass avocados in real time through videos. The main focus of the algorithm is based on analyzing the approximate dimensions of length and width of the fruit. The process begins by capturing the video in real time and then image processing techniques are applied to detect colors, generate contours, analyze dimensions and, based on these results, classify the avocado.

To ensure the accuracy of the algorithm, various tests were carried out in a controlled environment to avoid interference from external factors, such as light that could alter the colors captured by the camera. During the tests, 30 fruits were used and the measurements of the algorithm were compared with the results of a digital scale, following the criteria established by the Ecuadorian regulation NTE INEN 1755, 2nd Edition. The results showed an average percentage error of weight estimation of 0.06% and an algorithm efficiency of 96.97% in the Hass avocado grading tests.

Keywords: Artificial vision, Image Processing, Python, Avocado Classification, Hass Avocados, Weight Estimation.

ÍNDICE DE CONTENIDOS

CONSTANCIAS	III
CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR	IV
APROBACIÓN DEL COMITÉ CALIFICADOR	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
RESUMEN EJECUTIVO	VIII
ABSTRACT	IX
ÍNDICE DE CONTENIDOS.....	X
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABLAS	XVII
INTRODUCCIÓN	18
Planteamiento del Problema	18
Objetivos	20
Objetivo General.....	20
Objetivos Específicos.....	20
Justificación.....	21
Alcance.....	22
CAPÍTULO I: MARCO TEÓRICO	23

1.1. Antecedentes	23
1.1.1. Máquina Clasificadora de Aguacates Semiautomática por peso.....	23
1.1.2. Identificación de Madurez de la Granadilla con Tratamiento de Imágenes..	24
1.1.3. Clasificador de huevos	24
1.2. Fundamentos Teóricos.....	25
1.2.1. Aguacates	26
1.2.2. Variedades de Aguacates en Ecuador	26
1.2.2.1. Aguacate Fuerte.	26
1.2.2.2. Aguacate Hass.....	27
1.2.3. Clasificación de Aguacates en el Ecuador	28
1.2.4. NTE INEN 1755.....	28
1.2.5. Clasificadores de Aguacates.....	29
1.2.5.1. Manual.	29
1.2.5.2. UNISORTING.....	30
1.2.5.3. Visión Artificial.	31
1.2.6. Python	31
1.2.7. Visión Artificial.....	32
1.2.7.1. Sistema de Visión Artificial	32
1.2.7.2. Procesos en un Sistema de Visión Artificial	33
1.2.8. Componentes de un Sistema de Visión Artificial.....	34

1.2.8.1. Sistema de Procesamiento	34
1.2.8.2. Sistema de Iluminación	35
1.2.8.3. Sistema de Captación	38
CAPÍTULO II: MATERIALES Y MÉTODOS	40
2.1. Modelo de Investigación.....	40
2.1.1. Metodología de Desarrollo	40
2.2. Diseño de Investigación.....	40
2.2.1. Fase 1: Requerimientos	41
2.2.2. Fase 2: Diseño	41
2.2.3. Fase 3: Implementación	42
2.2.4. Fase 4: Verificación	42
2.2.5. Fase 5: Mantenimiento.....	42
CAPÍTULO III: RESULTADOS Y DISCUSIÓN	43
3.1. Levantamiento de Requerimientos.....	43
3.1.1. Requerimientos del Usuario	43
3.1.2. Requerimientos Técnicos	44
3.2. Análisis de Requerimientos	44
3.3. Propuesta de Algoritmo: Software	46
3.3.1. Software para Clasificación con Visión Artificial.....	46
3.3.2. Métodos de Clasificación con Visión Artificial.	47

3.4.	Propuesta de Algoritmo: Hardware.....	48
3.4.1.	Equipo de Computo	48
3.4.2.	Ambiente Controlado.....	48
3.4.3.	Iluminación.....	49
3.4.4.	Periférico para obtener imagen o video	50
3.5.	Desarrollo del Algoritmo	51
3.6.	Funcionamiento del Algoritmo	53
3.7.	Funcionamiento del Algoritmo: Software	53
3.7.1.	Detección de color	53
3.7.2.	Preprocesamiento y Segmentación de la Imagen	54
3.7.3.	Estimación de Peso	56
3.7.4.	Aplicación de etiquetas	58
3.8.	Funcionamiento del Algoritmo: Hardware	59
3.8.1.	Ambiente Controlado.....	59
3.8.2.	Iluminación.....	59
3.8.3.	Periférico de imagen	60
3.9.	Pruebas y Calibración.....	61
3.9.1.	Pruebas	61
3.9.2.	Identificación de Aguacates Hass.....	62
3.9.3.	Peso Estimado	63

3.9.4. Clasificación de Aguacates	63
3.9.5. Calibración	64
3.10. Resultados	65
3.10.1. Identificación de Aguacate Hass.	65
3.10.2. Análisis de Resultados de Identificación de Aguacate Hass	66
3.10.3. Peso Estimado.....	66
3.10.4. Análisis de Resultados de Peso Estimado	70
3.10.5. Clasificación de Aguacates.	70
3.10.6. Análisis de Resultados de Clasificación de Aguacates	71
CONCLUSIONES	74
RECOMENDACIONES	75
REFERENCIAS	76
ANEXOS.....	80

ÍNDICE DE FIGURAS

Figura 1 <i>Máquina clasificadora semiautomática de aguacates según el peso</i> [5].....	23
Figura 2 <i>Tratamiento de imágenes para identificar granadillas</i> [6].....	24
Figura 3 <i>Clasificador de huevos mediante visión artificial</i> [7].....	25
Figura 4 <i>Aguacate variedad Fuerte</i> [9].	27
Figura 5 <i>Aguacate variedad Hass</i> [9].	28
Figura 6 <i>Clasificación manual</i> [10].	29
Figura 7 <i>Maquinaria de UNISORTING</i> [11].....	30
Figura 8 <i>Arquitectura Python</i> [13].	31
Figura 9 <i>Ángulos de incidencia de la luz</i> [22].....	39
Figura 10 <i>Entorno de Anaconda Navigator</i>	46
Figura 11 <i>Entorno de Spyder</i>	47
Figura 12 <i>Estructura para ambiente controlado</i>	49
Figura 13 <i>Anillo de iluminación Led</i>	50
Figura 14 <i>Entorno de la aplicación móvil iVCam</i>	50
Figura 15 <i>Diagrama funcional del algoritmo</i>	51
Figura 16 <i>Diagrama de flujo del algoritmo de clasificación de aguacates Hass</i>	52
Figura 17 <i>Código de identificación de rangos de color</i>	53
Figura 18 <i>Array con los valores HSV del color a detectar</i>	54
Figura 19 <i>Conversión a HSV y detección de color</i>	54

Figura 20 <i>Aplicación de morfologías con kernel.</i>	55
Figura 21 <i>Aplicación de la detección de color y área.</i>	55
Figura 22 <i>Aplicación de morfologías con kernel.</i>	56
Figura 23 <i>Códigos de conversión y estimación del peso.</i>	57
Figura 24 <i>Comparativa de valores de peso.</i>	57
Figura 25 <i>Código para la generación de etiquetas informativas.</i>	58
Figura 26 <i>Ventana con el algoritmo funcional y presentación de etiquetas.</i>	58
Figura 27 <i>Diagrama de conexiones hardware.</i>	59
Figura 28 <i>Colocación y funcionamiento del aro de luz led en el ambiente controlado.</i> ...	60
Figura 29 <i>Entorno de iVCam conectado con el computador.</i>	60
Figura 30 <i>Funcionamiento del algoritmo de clasificación de aguacates.</i>	61
Figura 31 <i>Detección de Aguacates de la Variedad Hass.</i>	62
Figura 32 <i>Detección de aguacates y ausencia de detección en otras frutas.</i>	62
Figura 33 <i>Pesos estimado por el algoritmo y la medición en balanza digital.</i>	63
Figura 34 <i>Clasificación de aguacates con el algoritmo de clasificación.</i>	64
Figura 35 <i>Calculadora de coeficiente de Pearson.</i>	69

ÍNDICE DE TABLAS

Tabla 1 <i>Clasificación por masa unitaria</i> [4].	28
Tabla 2 <i>Fuentes de iluminación</i> [21].	35
Tabla 3 <i>Técnicas de iluminación</i> [21].	37
Tabla 4 <i>Tabla de requerimientos de usuario.</i>	43
Tabla 5 <i>Tabla de requerimientos de técnicos.</i>	44
Tabla 6 <i>Tabla de requerimientos analizados.</i>	44
Tabla 7 <i>Tabla de descripción de los requerimientos y sus elementos para cumplirlos.</i> ...	45
Tabla 8 <i>Características promedio de un equipo para correr visión artificial.</i>	48
Tabla 9 <i>Registro de la identificación de Aguacates de la Variantes Hass.</i>	65
Tabla 10 <i>Eficiencia de la detección de color.</i>	66
Tabla 11 <i>Registro de la comparativa de mediciones de peso.</i>	66
Tabla 12 <i>Errores porcentuales de las mediciones.</i>	67
Tabla 13 <i>Categorías del aguacate Hass según su masa.</i>	70
Tabla 14 <i>Categorías del aguacate Hass según su masa.</i>	71
Tabla 15 <i>Eficiencia de la clasificación de aguacates por su masa.</i>	72

INTRODUCCIÓN

En la parte introductoria de este trabajo se presenta la problemática y la propuesta de solución de manera clara y precisa que se desarrollará en el transcurso del proyecto.

Planteamiento del Problema

La producción de aguacates en Ecuador ha experimentado un crecimiento significativo en los últimos años, según un informe publicado por el Ministerio de Agricultura y Ganadería (MAG) en 2020. El país se ha convertido en uno de los principales productores de aguacates en la región gracias a una combinación de condiciones climáticas favorables, como la temperatura y la humedad, junto con la diversidad de suelos. En Ecuador, se cultivan dos variedades principales de aguacate: el Fuerte, que se destina principalmente al consumo nacional, y el Hass, que se exporta. Estos cultivos se encuentran distribuidos en aproximadamente 800 hectáreas ubicadas en diferentes provincias, incluyendo Carchi, Imbabura, Pichincha, Tungurahua, Azuay, Loja y Santa Elena. Durante el período de enero a mayo del año 2020, se exportaron alrededor de 232 toneladas de aguacates [1].

En el tema de la venta del aguacate el productor ecuatoriano sigue los lineamientos de tamaño y color que se encuentra en la norma ISO 1755. Esta normativa se instauró para clasificar y categorizar el aguacate para sus diferentes mercados [2]. Sin embargo, en la industria actual de producción y comercialización de esta fruta es frecuente que la selección inicial sea realizada por los fruticultores y a su vez, esta no sea precisa, lo que implica la necesidad de una segunda clasificación. Lamentablemente, este proceso se lleva a cabo manualmente, lo que resulta en una pérdida considerable de tiempo y depende en gran medida de las habilidades de las personas encargadas de la separación.

Dado que la clasificación manual puede llevar varias horas y los clasificadores suelen experimentar fatiga, es común que se cometan errores en los cálculos de costos de cada saquillo de aguacates. Esta situación no solo afecta a los comerciantes, quienes pueden incurrir en pérdidas monetarias, sino también a los agricultores, quienes también sufren las consecuencias financieras.

La implementación de un sistema de clasificación mediante visión artificial para los aguacates sería de gran ayuda tanto para los agricultores como para los comerciantes. Esta solución eficiente permitiría establecer de manera objetiva una clasificación correcta de la fruta en función de su tamaño y peso. Además, al eliminar la necesidad de clasificar manualmente, se reducirían los errores, los retrasos y los costos asociados, brindando beneficios económicos a todos los involucrados en la cadena de suministro de los aguacates.

Objetivos

Objetivo General

Diseñar un sistema para clasificar aguacates según su forma y tamaño.

Objetivos Específicos

- Definir los parámetros necesarios para la clasificación de aguacates.
- Determinar las herramientas computacionales que conformarán el programa de clasificación.
- Implementar el sistema de clasificación.
- Validar el sistema de clasificación.

Justificación

En los últimos años el Ecuador ha presentado un potente avance en el cultivo de aguacate para su exportación. En la región, Ecuador no destaca por la cantidad de hectáreas cultivadas de esta fruta sino por su calidad y el detalle en la clasificación y tratamiento del aguacate [3].

El presente estudio surge a partir de la recopilación de información, identificando una oportunidad para desarrollar un sistema de bajo costo que pueda automatizar las tareas de clasificación del aguacate Tipo Hass. Este sistema, con un nivel aceptable de precisión, busca apoyar a los agricultores y proporcionar información que actualmente no se recopila debido a la dificultad y el agotamiento físico y temporal que implica la clasificación manual. La implementación de este sistema permitiría realizar un conteo preciso en un período determinado, generando un valioso historial de información para la producción y comercialización, y estableciendo así el valor real del producto.

Alcance

El proyecto que se realizará tiene como objetivo generar un algoritmo de visión artificial de código abierto que permita clasificar los aguacates, uno a la vez, por su tamaño y forma. Este algoritmo será implementado en un ambiente controlado que permita garantizar una calidad óptima en cada una de las imágenes obtenidas. El medio controlado garantizará una posición e iluminación adecuada para la obtención de las imágenes mediante una cámara. Mediante pruebas y comparativas se validarán los resultados con la Norma Técnica Ecuatoriana NTE INEN 1755 Segunda Revisión 2015-10 [4].

CAPÍTULO I: MARCO TEÓRICO

1.1. Antecedentes

Los antecedentes permiten analizar trabajos relacionados para generar una primera perspectiva e idea del proyecto a realizarse extrayendo las partes de mayor relevancia y sintetizándolas en una nueva propuesta.

1.1.1. Máquina Clasificadora de Aguacates Semiautomática por peso

El sistema descrito en este trabajo detalla la aplicación de celdas de carga para clasificar los frutos basándose en varias categorías según su peso. Estas categorías se establecen al mercado destino, además de adicionar un contador para cada categoría. En la Figura 1, se visualiza el proceso semiautomático que empieza con la recolección de los aguacates cosechados en costales para ser colocados en la máquina clasificadora de aguacates por peso. Se enciende el equipo y finalmente, la máquina clasifica sin intervención humana los aguacates en canastas con fundas de diferentes colores como presenta en la figura.

Figura 1 Máquina clasificadora semiautomática de aguacates según el peso [5].

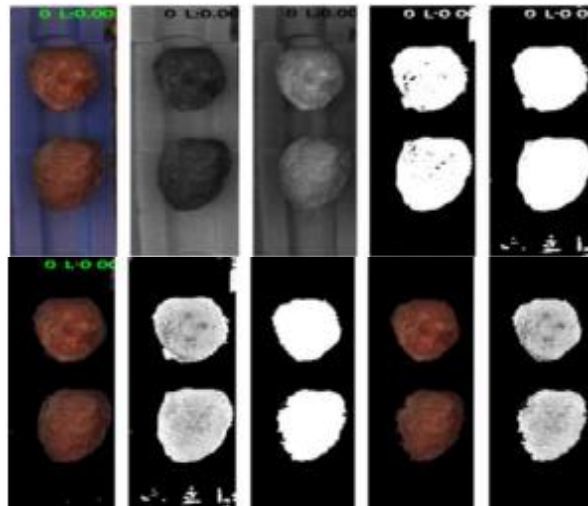


1.1.2. Identificación de Madurez de la Granadilla con Tratamiento de Imágenes

Con el incremento en la demanda de frutas exóticas, la producción de estos frutos ha experimentado un aumento constante. Sin embargo, en el proceso de selección, la identificación de frutas maduras aún se realiza manualmente, lo que conlleva a numerosos errores en el procedimiento.

Para abordar esta problemática, el autor desarrolló una herramienta computacional portátil (algoritmo de visión artificial) que utiliza técnicas de segmentación de imágenes para llevar a cabo diversos procesos de identificación de los frutos utilizando como punto principal la identificación del color del fruto con relación a su porcentaje de madurez. Este sistema ha sido programado en Python y se algunas pruebas de identificación de color y utilización de filtros se observa en la Figura 2.

Figura 2 Tratamiento de imágenes para identificar granadillas [6].

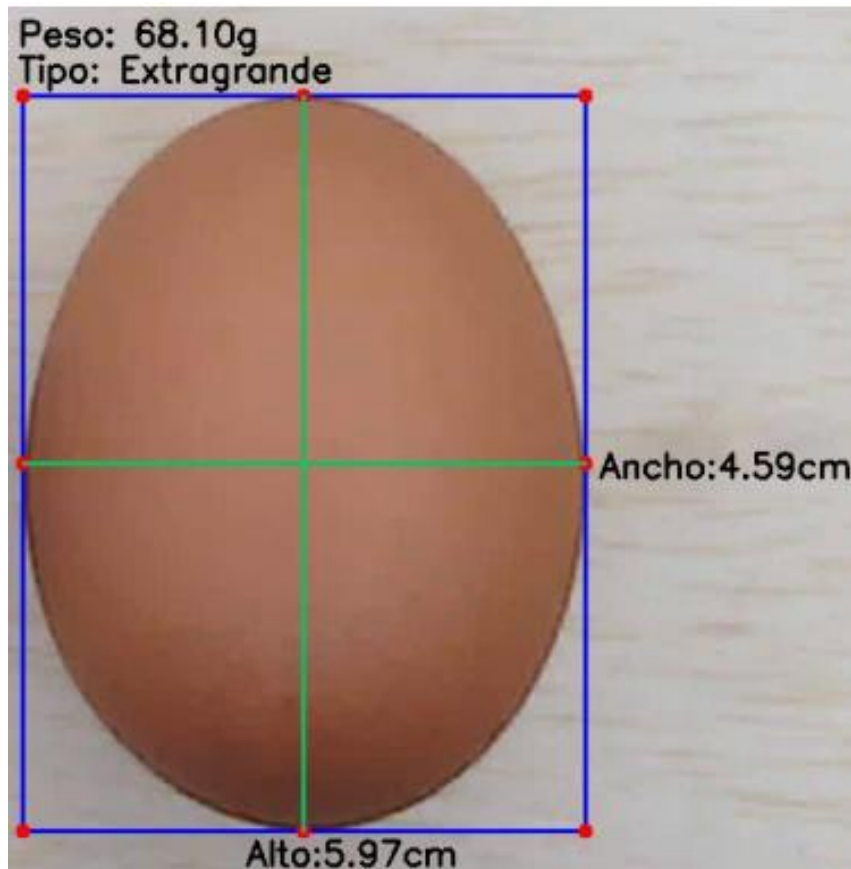


1.1.3. Clasificador de huevos

En este proyecto el autor utiliza las dos técnicas vistas en los numerales anteriores generando un sistema que parte de la identificación mediante visión artificial para generar una

clasificación mediante la aproximación de su peso. En el algoritmo analiza la forma, color y su densidad para genera las categorías mediante el peso. En la Figura 3, se observa el resultado que se observa en la pantalla al correr el algoritmo. En ella muestra un recuadro que estima las medidas como ancho y altura, su centroide, su peso estimado y su categoría.

Figura 3 Clasificador de huevos mediante visión artificial [7].



1.2. Fundamentos Teóricos

En la sección de fundamentos teóricos se agrupan todos los conocimientos fundamentales requeridos para llevar a cabo el desarrollo del proyecto, así como la presentación de las herramientas e información que describen el progreso y el cumplimiento de los objetivos establecidos.

1.2.1. Aguacates

El aguacate es considerado como uno de los cultivos más destacados en los valles Interandinos de Ecuador. Según datos del Ministerio de Agricultura y Ganadería (MAG) en 1997, se estimó que la superficie cultivada alcanzó las 3005 hectáreas, con rendimientos promedio de 14.996 kg por hectárea. Las provincias de Pichincha, Imbabura y Tungurahua destacaron por ser las de mayor extensión dedicadas al cultivo de esta fruta. El aguacate es ampliamente consumido por los ecuatorianos y su demanda en el mercado nacional ha ido en aumento, lo cual ha incentivado su producción [8].

1.2.2. Variedades de Aguacates en Ecuador

En Ecuador por lo general se habla de dos tipos de aguacates, la variedad Fuerte y la variedad Hass.

1.2.2.1. Aguacate Fuerte.

El aguacate Fuerte es reconocido como una de las variedades más deliciosas, Figura 4. Este aguacate de alta calidad se cosecha desde finales de otoño hasta la primavera. Se originó en México, específicamente en la zona de Puebla, y obtuvo su nombre debido a su capacidad para resistir un invierno particularmente severo, lo que lo distingue de otras variedades. Algunas de sus características son:

- Fruto: forma aperada y color verde claro
- Pulpa: excelente calidad
- Piel: fina y lisa, muy fácil de pelar.
- Hueso: tamaño medio
- Tamaño: 125- 500 gramos.
- Condiciones de transporte: buenas

Figura 4 *Aguacate variedad Fuerte* [9].



1.2.2.2. Aguacate Hass.

El aguacate Hass fue desarrollado en California a través de un proceso de selección natural entre las variedades Guatemalteca y Mejicana, y fue patentado por Rudolf G. Hass en 1920, Figura 5. Esta variedad se ha convertido en la favorita en la mayoría de los países. Debido a la diversidad de zonas de cultivo, en algunas áreas tempranas se puede encontrar desde otoño, mientras que en otras áreas más tardías su temporada se extiende hasta el verano. Algunas de sus características son:

- Fruto: forma ovoide, de color verde negruzco al madurar.
- Piel: medianamente gruesas y rugosa, flexible, fácil de pelar.
- Pulpa: excelente calidad y coloración.
- Hueso: tamaño medio.
- Tamaño: 125-350 gramos.
- Condiciones de transporte: excelentes

Figura 5 Aguacate variedad Hass [9].



1.2.3. Clasificación de Aguacates en el Ecuador

La clasificación de esta fruta para los diferentes mercados se base en clases o categorías las cuales se encuentran de talladas en la norma NTE INEN 1755 específicamente para el aguacate. Este proyecto se centrará en la clasificación del aguacate tipo Hass.

1.2.4. NTE INEN 1755

En la normativa INEN se describe todas las características físicas y químicas que un aguacate debe tener para entrar al proceso de clasificación, de no ser así este se descarta inmediatamente. A continuación, se presenta una tabla de clasificación del aguacate según su masa, Tabla 1.

Tabla 1 Clasificación por masa unitaria [4].

Escala de masa (m), en gramos	Grado (Número de frutos por caja de 5 kg)
$461 < m \leq 575$	8
$366 < m \leq 460$	10
$306 < m \leq 365$	12

$266 < m \leq 305$	14
$236 < m \leq 2655$	16
$211 < m \leq 235$	18
$191 < m \leq 210$	20
$156 < m \leq 190$	22
$146 < m \leq 155$	24
$136 < m \leq 145$	26
$125 < m \leq 135$	28
$80 < m < 125$	30

^a Variedad Hass exclusivamente. La diferencia entre el fruto más pequeño y el más grande no debe exceder los 25 g.

1.2.5. Clasificadores de Aguacates

Dentro del sector de producción existen algunas máquinas que realizan algún proceso de clasificación de aguacates, a continuación, se describen algunas.

1.2.5.1. Manual.

Por lo general la mayoría de los pequeños productores realizan los procesos de clasificación de manera manual, utilizando su perspectiva, siendo un estándar no muy fiable y susceptible para tener diferentes interpretaciones. Además, se estar sujeta al esfuerzo humano, Figura 6.

Figura 6 *Clasificación manual* [10].



Nota: En la imagen se observa al personal realizando una inspección y colocación correcta de aguacate.

1.2.5.2. UNISORTING.

La empresa UNISORTING diseña y vende maquinaria para el tratado del aguacate. En estas máquinas que constituyen una línea de producción se realizan los procesos de limpieza, cepillado, clasificación. Por la magnitud, de la maquinaria, por defecto, se estima que este tipo de equipo se emplean únicamente en industrias de gran magnitud para poder cubrir el tamaño y costo de todo este equipo. En la Figura 7 se observa un ejemplo de los equipos que esta empresa ofrece.

Figura 7 *Maquinaria de UNISORTING* [11].



Nota: Máquina automática de clasificación de aguacates de la empresa UNISORTING.

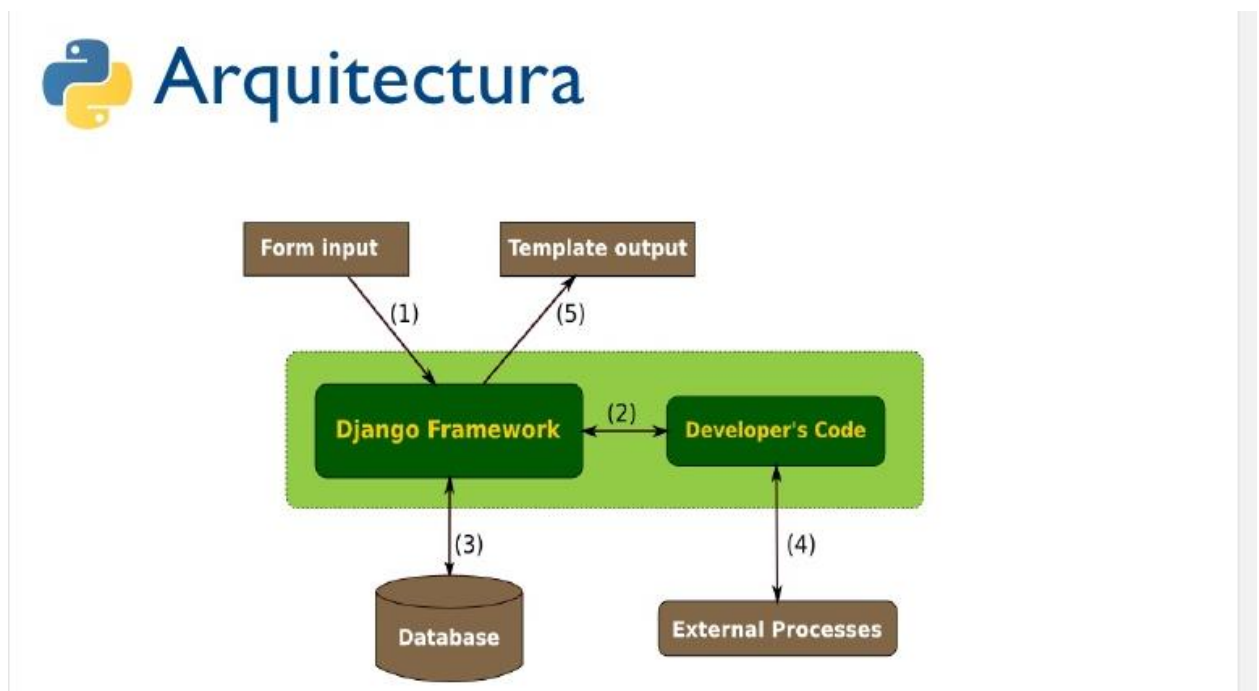
1.2.5.3. Visión Artificial.

El objetivo de este proyecto es generar un algoritmo que permita potenciar el proceso de clasificación de los pequeños productos, y se instaure como una alternativa de bajo costo. Pero a su vez pueda tener características de maquinaria de costos elevador destinado en su mayoría a grandes industrias.

1.2.6. Python

Es un lenguaje de programación de alto nivel orientado a objetos que se utiliza comúnmente en el diseño web y desarrollo de aplicaciones. Python destaca por su semántica integrada dinámica, lo que lo hace atractivo y fácil de aprender. Cuenta con una amplia gama de librerías y su arquitectura se representa en la Figura 8 [12]. En el proceso de programación y generación del algoritmo, se emplean diversas librerías que permiten simplificar y optimizar el código. Algunas de estas librerías son:

Figura 8 *Arquitectura Python* [13].



Matplotlib: se enfoca en la creación de gráficos y diagramas. En el ámbito de la visión artificial, es una herramienta valiosa para dibujar, señalar y seleccionar objetos en imágenes o videos analizados [14].

Numpy: es una librería diseñada especialmente para el análisis de datos y el cálculo numérico a gran escala. A diferencia de otras librerías, Numpy permite realizar operaciones en arrays hasta 50 veces más rápido, siendo ideal para manipular vectores y matrices de dimensiones amplias [14].

OpenCV: es una librería orientada al procesamiento de imágenes en tiempo real, particularmente en el ámbito de la visión por computadora [15].

1.2.7. Visión Artificial

Los estudios relacionados con la visión artificial se centran en la simulación de la capacidad visual humana utilizando computadoras. Este enfoque implica capturar imágenes o videos con el objetivo de analizar la información y las características de los objetos, con el fin de comprender los factores involucrados en el objeto de estudio.

Para lograr este propósito, se emplean técnicas que permiten adquirir, procesar, analizar y comprender las imágenes del mundo real, de manera que la información pueda ser interpretada por el ordenador. Esto permite que la computadora actúe de manera similar a una persona en una situación específica [16].

1.2.7.1. Sistema de Visión Artificial

Según [13], un sistema de visión artificial consiste en la combinación de elementos que se unen con el objetivo de obtener información del mundo físico a través de imágenes

bidimensionales. Para lograr esto, se utiliza un ordenador que, mediante un algoritmo específico, es capaz de reconocer, analizar y comprender la escena para la cual ha sido programado. En otras palabras, el ordenador procesa las imágenes con el fin de extraer información relevante.

1.2.7.2. Procesos en un Sistema de Visión Artificial

Cada aplicación de visión artificial presenta características específicas debido a que el reconocimiento de imágenes se encuentra directamente relacionado con el objetivo que se pretende alcanzar. Aunque no hay un conjunto de pasos definidos para seguir, existen una serie de etapas comunes y habituales en el proceso de visión artificial, que se describen a continuación.

La primera etapa es la adquisición de imágenes, considerada primordial ya que se busca obtener una imagen adecuada para el análisis actual y las etapas siguientes. La elección del método de adquisición depende del objeto de interés y del propósito del estudio [17].

Posteriormente, se lleva a cabo el preprocesamiento de imágenes, donde se aplican técnicas para mejorar la calidad de la imagen, eliminando imperfecciones, resaltando características, mejorando la intensidad, entre otros aspectos.

La etapa de segmentación permite dividir la imagen capturada en diferentes partes, con el fin de extraer las características necesarias. Una segmentación adecuada garantiza una solución correcta y confiable en el sistema desarrollado [18].

Luego, se realiza la extracción de características, donde se busca obtener de manera precisa las características deseadas para su respectivo análisis. Los datos extraídos suelen basarse en aspectos como color, distancias, áreas, ángulos, contornos, entre otros.

Finalmente, se lleva a cabo el análisis y reconocimiento de la imagen, donde los sistemas de visión artificial tienen diferentes objetivos. Estos incluyen la identificación de objetos presentes en la imagen, la clasificación de objetos según sus características, las mediciones de objetos y el conteo de objetos en la imagen [19][20].

1.2.8. Componentes de un Sistema de Visión Artificial

Un sistema de visión artificial se puede segmentar en tres partes principales las cuales se describen a continuación.

1.2.8.1. Sistema de Procesamiento

La visión artificial se encarga de extraer datos e información de imágenes o videos, pero antes de eso, se realiza un procesamiento para obtener la información necesaria de manera más sencilla y eficiente. A continuación, se presentan algunos de los algoritmos más utilizados:

Escala de grises: En ocasiones, el color de una imagen no facilita la interacción necesaria para obtener la información requerida. Transformar una imagen a escala de grises permite ahorrar información y optimizar los cálculos y procesos en el análisis de imágenes.

Ajuste del contraste: El ajuste del contraste tiene como objetivo aumentar o disminuir la intensidad general de los píxeles de una imagen. Esto se logra manipulando las partes oscuras y brillantes.

Segmentación: La segmentación implica dividir una imagen en regiones o segmentos, resaltando áreas específicas para un análisis especializado. Existen diferentes tipos de segmentación, como la segmentación por color, texturas, semántica y Superpíxel.

Umbralización: La umbralización consiste en asignar un valor umbral y, mediante una comparación, cambiar el píxel a blanco o negro. Se utilizan diferentes tipos de umbralización.

Detección de bordes: La detección de bordes permite identificar las transiciones entre diferentes regiones en una imagen, proporcionando información sobre las fronteras de objetos, colores o texturas.

Detección de objetos: La detección de objetos utilizando la técnica YOLO se basa en el uso de redes neuronales. Se crea un conjunto de datos (Dataset) y se realiza la detección mediante programación en Python.

1.2.8.2. Sistema de Iluminación

Las fuentes de iluminación desempeñan un papel fundamental en el control de cómo las muestras son capturadas por el sensor o la cámara, con el fin de garantizar que se cumplan los requisitos de adquisición de información [21]. En la Tabla 2 se presentan diversas fuentes de iluminación que se utilizan en este contexto.

Tabla 2 Fuentes de iluminación [21].

Tipo de Fuente de Luz	Ventajas	Desventajas
Halógenas	Alta luminosidad ajustable	No es posible emplearla como luz estroboscópica. Caliente. Se degrada con el tiempo.
Incandescencia	Bajo costo	No es posible emplearla como luz estroboscópica. Alta temperatura. Alta degradación.

Fluorescentes	Variedad de formas y tamaños. Minimiza los reflejos.	No es posible emplearla como luz estroboscópica. Parpadeo. Alta dificultad para atenuar. Alta degradación.
Laser	Es posible emplearla como luz estroboscópica. Útil en 3D.	Costo elevado. Textura gránulos. Alta fragilidad. Atenta con la salud ocular para los trabajadores.
Xenón	Es posible emplearla como luz estroboscópica. Se puede emplear con alta intensidad en periodos cortos (flash).	Alta degradación.
LED	Se puede emplear con alta intensidad en periodos largos (10.000 hora o más) Es posible emplearla como luz estroboscópica. Posee un bajo consumo	Costo elevado.

Con el objetivo de capturar detalles específicos de una imagen a analizar, se emplean diferentes técnicas de iluminación que involucran la variación del ángulo de incidencia, la posición de la iluminación e incluso la fuente de iluminación utilizada [21]. En la Tabla 3 se presentan una serie de técnicas de iluminación utilizadas con este propósito.

Tabla 3 *Técnicas de iluminación* [21].

Iluminación	Aplicaciones	Ventajas	Inconvenientes
Frontal	Permite analizar superficies con pocos reflejos: papel, tela. Detectar de marcas. Detectar de colores y caracteres.	Permite eliminar sombras, se puede utilizar a grandes distancias cámara/objeto.	Altos reflejos sobre superficies reflectantes.
Lateral	Permite resaltar bordes, rayas y fisuras con una dirección determinada.	Permite resaltar relieves muy pequeños, resultando una sombra muy definida.	La luz con ángulos respecto a la horizontal producirá relieves y contornos en las piezas.
Campo Oscuro	Permite resaltar incrustaciones y códigos alfanuméricos con poco contraste en metal sobre metal o gris sobre gris.	Destaca los detalles en superficies con muy poco contraste.	No es posible reconocer detalles superficies que absorban la luz.
Contraste	Indicado para inspeccionar la silueta del objeto.	Permite inspeccionar siluetas e impurezas.	No permite reconocer detalles superficiales del objeto, códigos, inscripciones, etc.
Axial difusa	Permite inspeccionar superficies planas reflectantes.	Permite inspeccionar materiales altamente reflectantes.	No puede reconocer relieves.
Difusa tipo domo	Aplicaciones en inspecciones con uso médico, espejos, compact disk, latas, etc.	Aplicaciones para eliminar sombras y minimizar arrugas, polvos y relieves	Alto costo.
Láser	Ajuste en procesos de corte. Análisis de profundidad de objetos.	No le influye la iluminación.	Para conseguir una línea es necesario lentes cilíndricos.

1.2.8.3. Sistema de Captación

La cámara es un dispositivo que captura información en forma de imágenes digitales al ser sensible a la luz. Al elegir una cámara, es importante considerar aspectos relacionados con el entorno, la interacción y la variación de la iluminación.

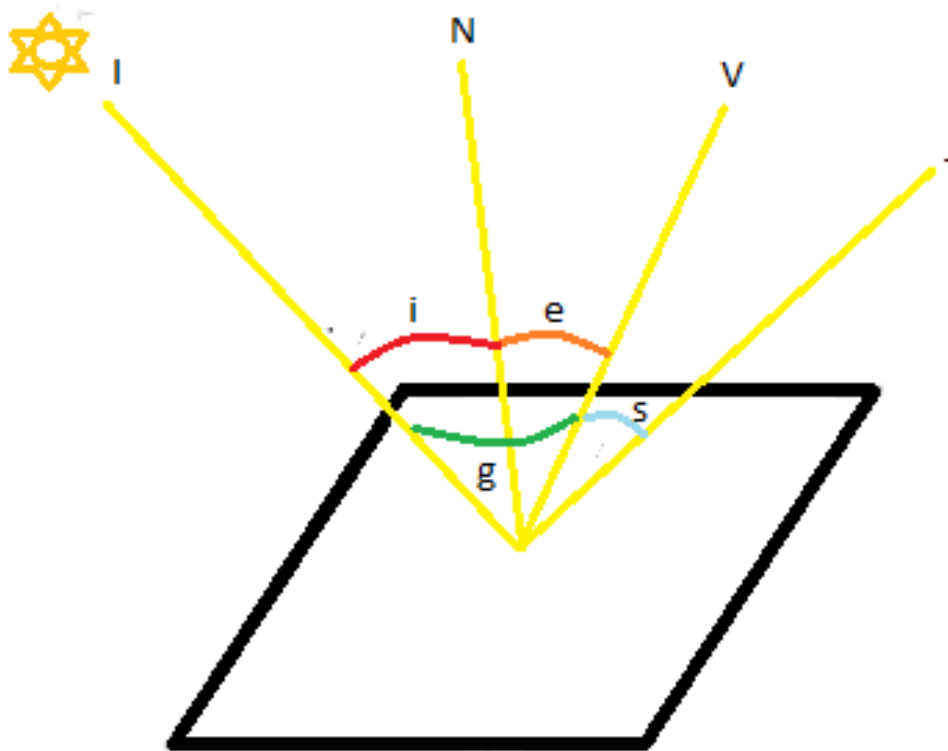
Obtención de la imagen: La captura de una fotografía digital se puede describir como un proceso en el que un sensor fotosensible captura la luz en una cámara oscura. El sensor está compuesto por unidades fotosensibles que detectan la luz en tres componentes de color: rojo (R), verde (G) y azul (B). Cada píxel de la imagen se completa con una combinación de colores basada en la luz captada por el sensor.

El color: Para el ojo humano, todos los colores se perciben como combinaciones de tres colores primarios: azul, verde y rojo. La mezcla adecuada de estos colores resulta en el color blanco. En las imágenes digitales, los colores se representan en la gama RGB, donde el matiz, la saturación y la intensidad se analizan en una escala de 0 a 255. Los componentes rojo (red), verde (green) y azul (blue) se pueden visualizar en un sistema tridimensional.

Reflejo: Cuando observamos un objeto, nuestra visión humana puede mantener la precisión de la imagen a pesar de las variaciones en la luminosidad. Todas las superficies tienen la capacidad de reflejar la luz según la incidencia de esta sobre el objeto. Las superficies pueden ser brillantes, mate, translúcidas u opacas. Por lo tanto, es importante conocer los ángulos de incidencia de la luz para colocar adecuadamente la iluminación sobre el objeto y obtener la mayor cantidad de información con la calidad necesaria [22].

En la Figura 9, se presenta cada uno de los ángulos de incidencia de un haz de luz el cual es importante para definir su colocación en el ambiente controlado.

Figura 9 *Ángulos de incidencia de la luz* [22].



- **s** representa el ángulo no especular.
- **g** representa el ángulo de fase.
- **N** es la normal de la superficie.
- **e** es el ángulo de salida.
- **V** es la dirección de observación.
- **J** es la dirección de reflexión especular.
- **i** es el ángulo de incidencia.
- **I** es la dirección de iluminación.

CAPÍTULO II: MATERIALES Y MÉTODOS

Este capítulo proporciona una descripción detallada de la metodología empleada para desarrollar el diseño del algoritmo de visión artificial enfocado en la clasificación del aguacate tipo Hass. Además, se explica el procedimiento para verificar los objetivos específicos establecidos en este documento.

2.1. Modelo de Investigación

El presente proyecto utiliza diferentes métodos de investigación para abordar un problema que surge en el proceso de selección de botones de rosas maduras para su posterior recorte y transporte. La investigación documental se lleva a cabo de manera sistemática, revisando libros, catálogos, entre otros recursos, con el objetivo de recopilar la información necesaria para establecer las bases teóricas del proyecto. En cuanto a la aplicación práctica del proyecto, se enfoca en la implementación y pruebas en el campo [23].

2.1.1. Metodología de Desarrollo

En términos de la metodología de investigación, es importante mencionar y seguir un enfoque de desarrollo como el modelo de cascada, que consta de cinco fases secuenciales que describen las etapas y actividades para llevar a cabo el proyecto. Estas fases son: análisis de requerimientos, diseño, implementación, verificación y mantenimiento [24]. Cada una de estas etapas implica realizar actividades que deben ser completadas antes de pasar a la siguiente.

2.2. Diseño de Investigación

Tomando en cuenta lo expuesto previamente, es posible organizar las etapas según los objetivos establecidos y las actividades correspondientes para cada proceso.

2.2.1. Fase 1: Requerimientos

En esta primera fase se definen los requerimientos en base al tipo de aguacate orientada con la ayuda de la normativa INEN. Para la obtención de información se sintetiza datos encontrados mediante investigación y entrevistas a beneficiarios o técnicos en el cultivo y procesamiento del aguacate tipo Hass.

Actividad 1: Con la información revisada previamente se constituye una entrevista concreta buscando identificar los requerimientos del proyecto.

Actividad 2: El análisis de la información recolectada para sintetizarlos en requerimientos.

Actividad 3: Se realiza un proceso de selección e identificación de procesos para determinar las especificaciones y los parámetros que permitan identificar y seleccionar los aguacates tipo Hass.

Actividad 4: Se plantea la propuesta de un algoritmo de visión artificial que tenga como funcionalidad principal la identificación y clasificar huevos por el peso aproximado.

2.2.2. Fase 2: Diseño

A partir de la primera fases se continua con las actividades en el diseño.

Actividad 1: Planteamiento de la propuesta del algoritmo con sus funcionalidades y cumplimiento de requerimientos.

Actividad 2: Selección de software, librerías y métodos para desarrollar el algoritmo.

Actividad 3: Selección del hardware para obtención de las imágenes con la mejor calidad para su procesamiento.

Actividad 4: Desarrollo del algoritmo mediante la visión artificial.

2.2.3. Fase 3: Implementación

Actividad 1: Prueba y selección de librerías y métodos de procesamiento de imágenes en tiempo real.

Actividad 2: Constitución del algoritmo mediante visión artificial con el procesamiento de imágenes en tiempo real.

2.2.4. Fase 4: Verificación

La penúltima fase se centra en poner a prueba el trabajo realizado, en esta fase se verifica y se realiza cambios en caso de ser necesarios.

Actividad 1: Ajuste de los rangos de colores en la escala HSV.

Actividad 2: Ajuste de filtros y ajustes de imagen.

Actividad 3: Ajuste y calibración de la escala en relación entre cm y píxel.

Actividad 4: Pruebas de funcionamiento.

Actividad 5: Comparativa de magnitudes con elementos de medición.

2.2.5. Fase 5: Mantenimiento

La última fase en resumen contiene a las actividades de calibración final y mantenimiento o ajustes que necesite el algoritmo después de un tiempo de funcionamiento o ajuste nuevos bajo nuevas circunstancias o condiciones.

CAPÍTULO III: RESULTADOS Y DISCUSIÓN

En el tercer capítulo, se procede a avanzar desde el diseño previo hacia la materialización de un algoritmo funcional. Se verifica el funcionamiento del software a través de pruebas bajo diversas condiciones, se realizan correcciones y ajustes finales para asegurar su total operatividad.

3.1. Levantamiento de Requerimientos

Para el levantamiento de los requerimientos se procedió a analizar la información presentada en la NTE INEN 1755 acerca de la clasificación del aguacate para la exportación. Además, mediante una entrevista a un productor de aguacates se extrae la información para estructurar los siguientes requerimientos.

3.1.1. *Requerimientos del Usuario*

Mediante la entrevista se puede sintetizar la Tabla 4 con los requerimientos de quien sería uno de los beneficiarios para el desarrollo de este algoritmo.

Tabla 4 *Tabla de requerimientos de usuario.*

Requerimientos	Descripción
Precisión	Alta precisión
Procesamiento	Rápidos en la clasificación
Facilidad de uso	Sencillez en usar y calibrar
Economía	Su construcción y realización tengo un valor accesible
Adaptabilidad	Sea utilizable en diferentes lugares

3.1.2. *Requerimientos Técnicos*

Obtenidos los requerimientos de usuario por parte de productor se genera la equivalencia con aspectos técnicos como se observa en la Tabla 5.

Tabla 5 *Tabla de requerimientos de técnicos.*

Requerimientos	Descripción
Procesamiento	Capacidad Computacional media-alta
Software de programación	Libre de alto nivel
Precisión	Alta precisión
Luminosidad	Ambiente controlado
Adquisición de video	Resolución alta del periférico
Ambiente controlado	Económico

3.2. **Análisis de Requerimientos**

A partir de la información obtenida y las tablas generadas con anterioridad se procede a realizar la Tabla 6, la cual resume de manera más clara los requisitos.

Tabla 6 *Tabla de requerimientos analizados.*

Nomenclatura	Requerimientos	Elementos que permite cumplir el requerimiento
Re1	Lenguaje de alto nivel	Python
Re2	Precisión	Visión Artificial OpenCV
Re3	Procesamiento	Computo medio-alto

Re4	Adquisición de imagen	Cámara de resolución alta
Re5	Luminosidad	Iluminación led ajustable
Re6	Medio controlado	Estructura con fondo

Mediante la Tabla se realiza una nueva tabla que permite visualizar las ventajas y desventajas de emplear los diferentes elementos para cumplir los requerimientos, Tabla 7.

Tabla 7 *Tabla de descripción de los requerimientos y sus elementos para cumplirlos.*

Elemento que permite cumplir el requerimiento	Ventajas	Desventajas
	Lenguaje de alto nivel	
Lenguaje Python	Lenguaje de código abierto Módulos y Librerías libres	Necesita alto computo Ralentiza los equipos
	Enfoque automático	Tamaño y resolución
iVCam	Resolución media-alta del celular	variable Conexión a PC
	Luminosidad y color ajustable	Temperatura alta
Aro luces leds	Portabilidad	Costo
	Económico	Acabados y presentación
Ambiente controlado	Materiales de casa	variable

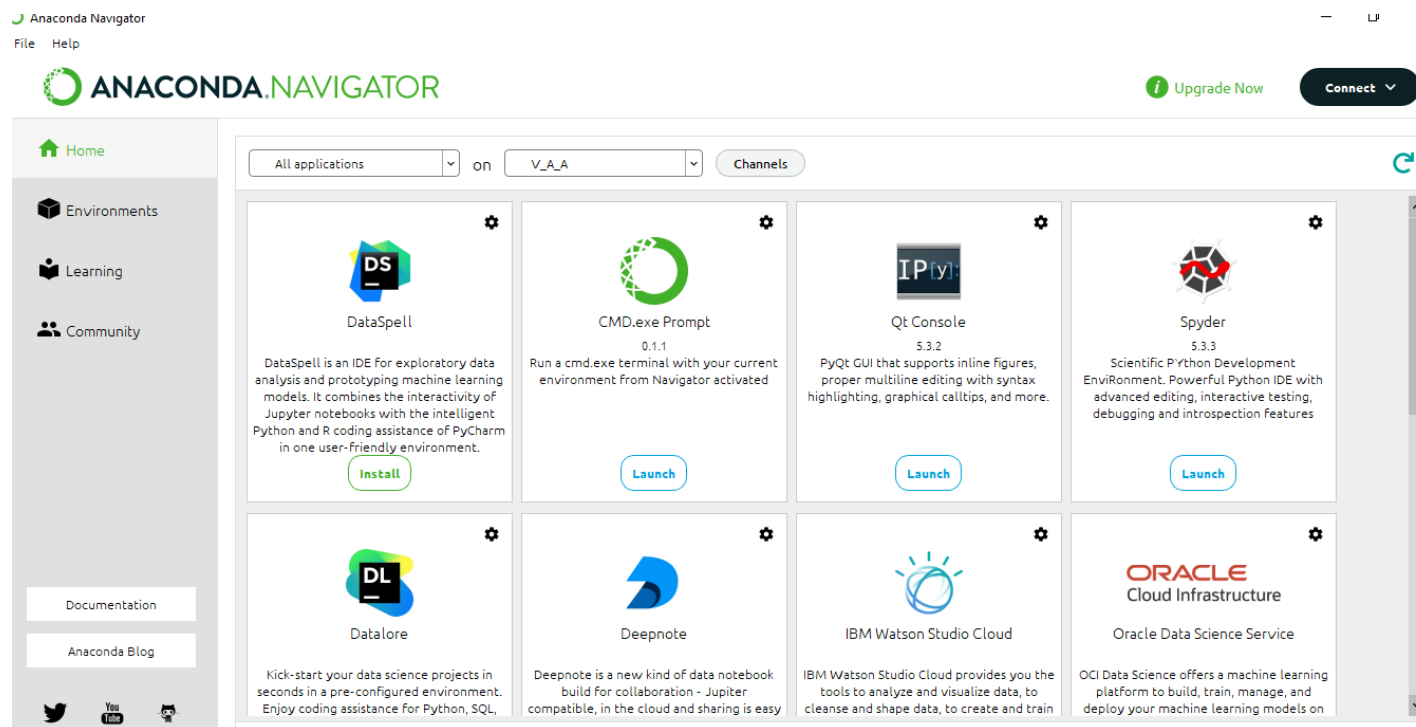
3.3. Propuesta de Algoritmo: Software

Como punto de partida para la propuesta del algoritmo se detalla el programa y el método que se aplica para el desarrollo del proyecto.

3.3.1. Software para Clasificación con Visión Artificial

Para el tema del software que se puede utilizar para generar y probar el algoritmo es importante conocer que el lenguaje de programación Python puede ser ejecutado de manera nativa en sistemas operativos basados en Linux o utilizar librerías, entornos virtuales o ecosistemas que puedan correr este lenguaje de programación.

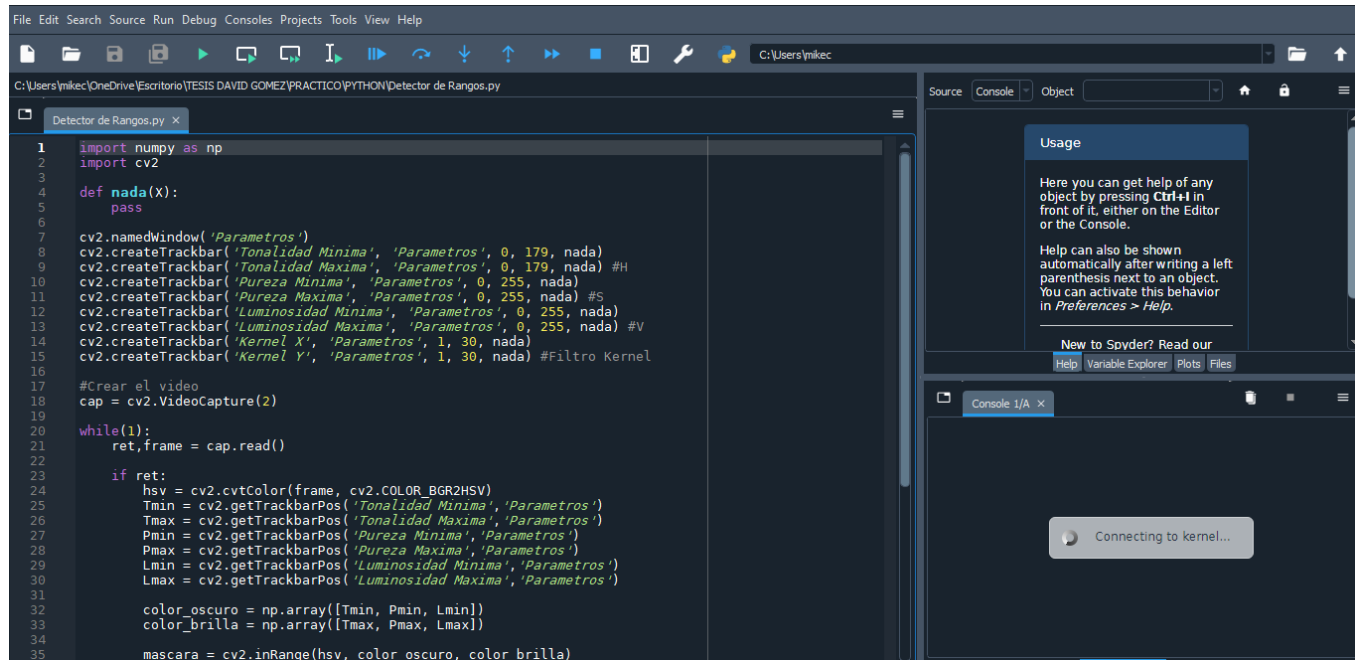
Figura 10 Entorno de Anaconda Navigator.



En el SO de Windows se emplea la aplicación Anaconda Navigator, Figura 10, el cual es una aplicación que permite gestionar diferentes ecosistemas entre ellos Spyder. Este editor y

ejecutor de Python permite utilizar las librerías previamente instaladas y trabajar infinidad de proyectos en Python, Figura 11.

Figura 11 Entorno de Spyder.



3.3.2. Métodos de Clasificación con Visión Artificial.

Existen diferentes métodos para poder generar un código de clasificación de aguacates. En este proyecto se usará el tratamiento de imágenes, el cual, se centra en utilizar imágenes o videos (en tiempo real), aplicar diferentes filtros y ajustes a las imágenes para identificar ciertas características.

En este caso el algoritmo tendrá como base dos características físicas del aguacate: su color y su área (tamaño). A partir de esto, se emplean filtro de detección de colores los cuales solo funcionarán con los colores del aguacate tipo Hass y después se realizarán los siguientes cálculos.

3.4. Propuesta de Algoritmo: Hardware

En esta sección se describe el hardware necesario para dar vida y funcionamiento al algoritmo de visión artificial para la clasificación de aguacates.

3.4.1. Equipo de Computo

Para la programación es necesario un equipo de cómputo de características medias-altas para evitar fallos o que el sistema deje de responder. En la Tabla 8 se observan algunas características que se puede tener en cuenta para poder programar y ejecutar un código de visión artificial con Python.

Tabla 8 Características promedio de un equipo para correr visión artificial.

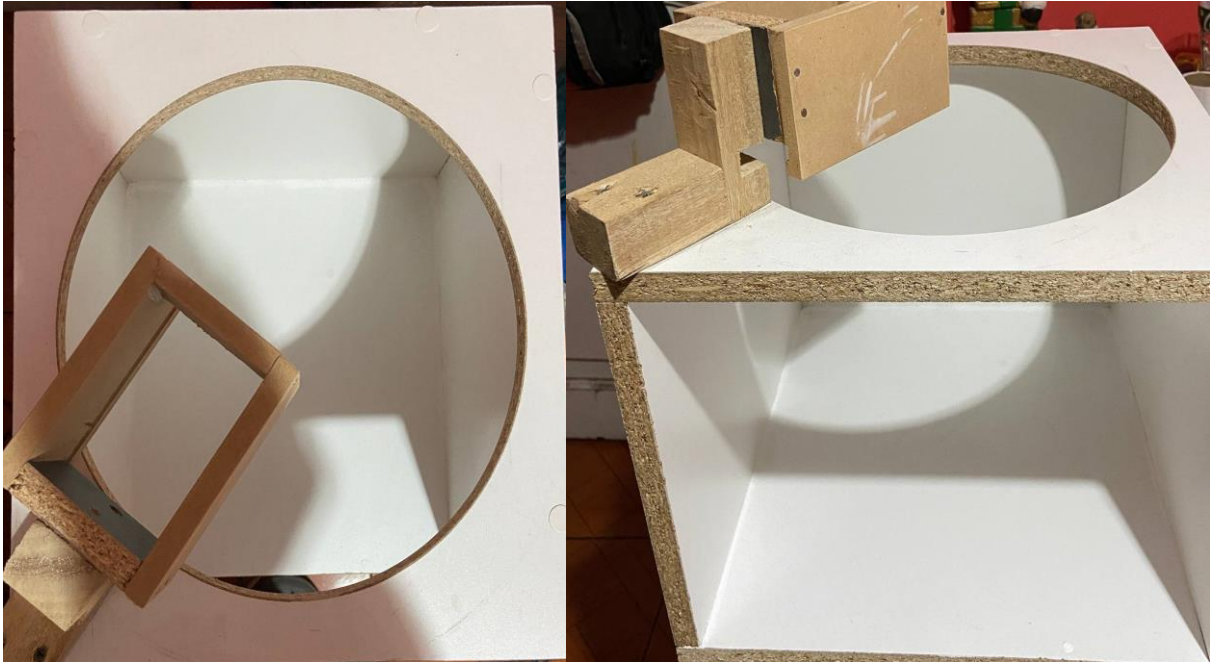
ESPECIFICACIONES DEL DISPOSITIVO	
Procesador	Intel(R) Core (TM) i5 o mayor CPU @ 2.80GHz 2.80 GHz o mayor
RAM instalada	8.0 GB o mayor
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
OS	Windows 10 Pro

3.4.2. Ambiente Controlado

Para asegurar las mejores condiciones de iluminación y posición al momento de clasificar los aguacates, se construye una estación o caja cuadrada de las dimensiones 20cm x 20 cm x 20

cm la cual está constituida de madera para evitar costos elevados. Adicionalmente en la parte superior cuenta con un acople específico para colocar la cámara y la iluminación, Figura 12.

Figura 12 Estructura para ambiente controlado.



3.4.3. Iluminación

En el apartado de la iluminación existe gran cantidad de elementos. Sin embargo, se emplea una iluminación blanca de bajo consumo que a su vez permita ser ajustable. Con estas necesidades la iluminación Led permite cumplir todas estas prestaciones. En la Figura 13, se observa un anillo luminoso con una conexión mediante USB, un diámetro de 12 pulgadas y un control de intensidad de iluminación. Por lo general, un aro de luz presenta una eficiencia lumínica de 80 a 100 lúmenes por vatio. Según [25], 70 lúmenes por vatio es el valor más común. Si el aro de luz utilizado tiene una potencia de 12W, su potencia lumínica es aproximadamente 840 lm.

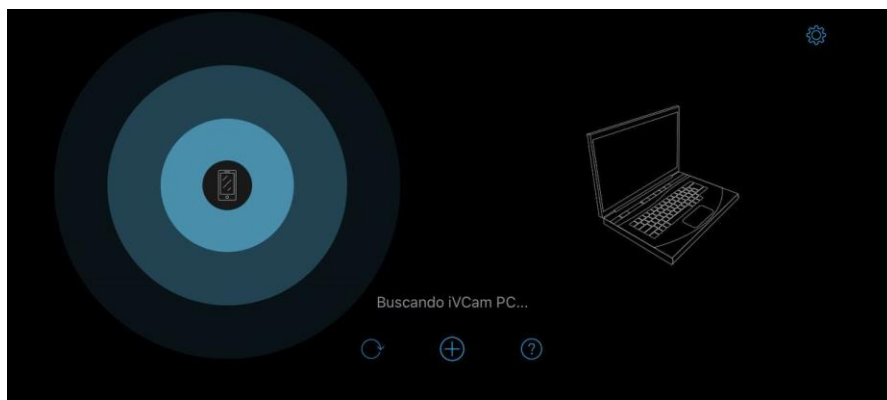
Figura 13 *Anillo de iluminación Led.*



3.4.4. Periférico para obtener imagen o video

La cámara será uno de los elementos esenciales para poner a funcionar el algoritmo de clasificación. Para esto se recomienda una cámara con altas prestaciones como: resolución 720 a 1080 píxeles, mayor a 8 megapíxeles, autoenfoco. Sin embargo, comercialmente este tipo de equipos tienen un costo relativamente elevado. Partiendo del requerimiento de bajo costo se buscó una alternativa mejor empleando la aplicación iVCam, Figura 14, permitiendo enlazar la cámara de los dispositivos móviles con el computador y a su vez permitiendo utilizar la versatilidad de las cámaras de celular que actualmente la mayoría de los equipos tiene media-altas prestaciones.

Figura 14 *Entorno de la aplicación móvil iVCam.*



3.5. Desarrollo del Algoritmo

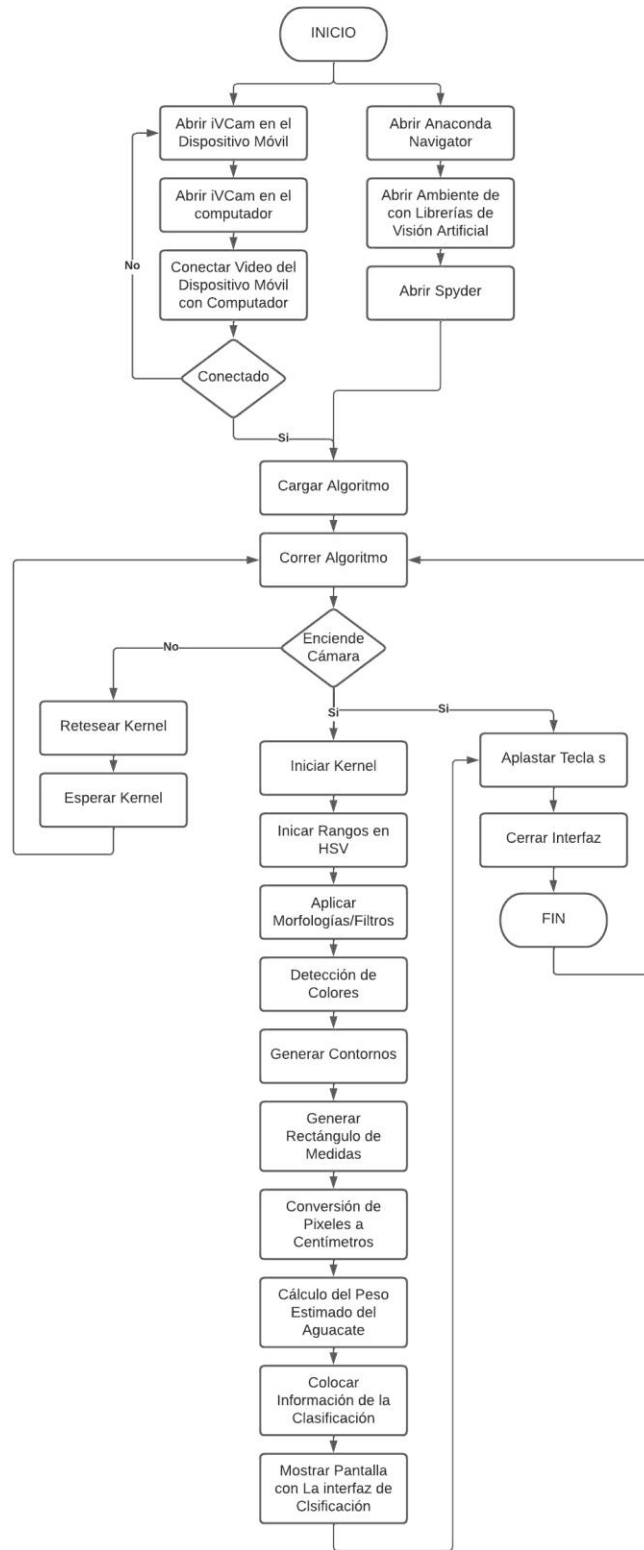
El desarrollo del algoritmo comienza desde la conceptualización de las funciones y las interacciones de los procesos que conlleva el funcionamiento del sistema. En la Figura 15, se presenta un diagrama funcional que resume el proceso general del algoritmo con sus entradas y salidas.

Figura 15 *Diagrama funcional del algoritmo.*



A continuación, es posible detallar de manera más precisa todos los procesos que realizará el algoritmo de manera secuencial. Para realizar esta contextualización es recomendable utilizar herramientas que faciliten estructuras y presentar toda esta información de la mejor manera, en la Figura 16 se observa un diagrama de flujo con los procesos del algoritmo de clasificación.

Figura 16 Diagrama de flujo del algoritmo de clasificación de aguacates Hass.



3.6. Funcionamiento del Algoritmo

Esta sección presenta de manera segmentada los procesos y acciones que se ejecutan en la parte de software y hardware de manera independiente y su trabajo como un sistema integral. Es importante detallar estos procesos para comprender el funcionamiento del algoritmo.

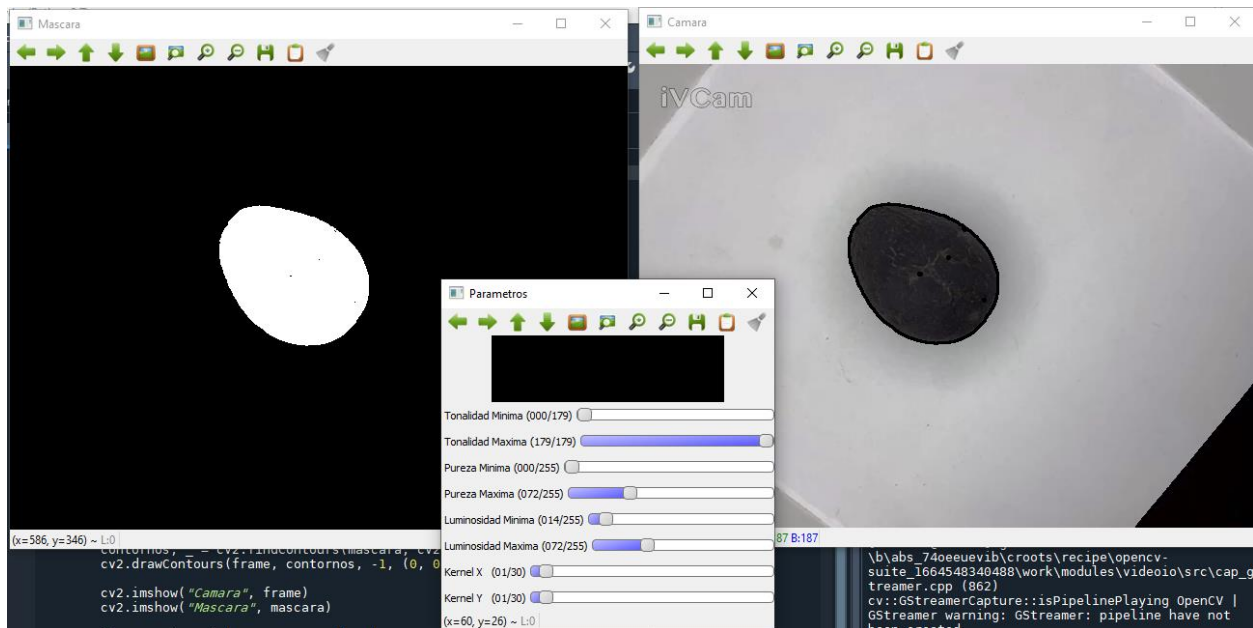
3.7. Funcionamiento del Algoritmo: Software

La clasificación del algoritmo se divide en algunos subprocesos:

3.7.1. Detección de color

Para el caso del aguacate tipo Hass predomina dos tonalidades del color verde para ser detectado. Para que el algoritmo pueda detectar cierto color o colores necesita de valores correspondientes al color en formato HSV. Para esto se utilizó un código adicional el cual permite detectar el rango de color de una manera mucho más exacta, se ajustan los sliders de manera manual hasta ajustarlo al color necesario, Figura 17.

Figura 17 Código de identificación de rangos de color.



Listo los valores en HSV se genera una máscara de color el cual permitirá al algoritmo identificar el área del color necesario. Para esto se genera una array con el límite inferior y superior, Figura 18.

Figura 18 Array con los valores HSV del color a detectar.

```
"Límite Inferior de Color"  
ri = np.array([0, 0, 14],np.uint8)  
"Límite Superior de Color"  
rs = np.array([179, 72, 72],np.uint8)
```

Dentro del código, se utiliza unas líneas de la Figura 19 donde se convierte la imagen de colores RGB a HSV. A continuación, con los valores HSV obtenidos se aplica la función cv2.inRange para diferenciar la parte de color que se necesita de los demás colores.

Figura 19 Conversión a HSV y detección de color.

```
while True:  
    "Lectura de la Cámara"  
    ret,frame = cap.read()  
    if ret==True:  
        "Conversión BGR a HSV"  
        frameHSV = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)  
        "Máscara de Color para Identificar el Aguacate"  
        mask = cv2.inRange(frameHSV,ri,rs)
```

3.7.2. Preprocesamiento y Segmentación de la Imagen

Dentro del procesamiento se emplean diferentes líneas de programación para mejorar la detección aplicando Morfologías con un kernel. Estas Morfologías permiten eliminar zonas pequeñas detectadas, unir zonas que se encuentran separadas ligeramente y que permita generar mejor un área, Figura 20.

Figura 20 Aplicación de morfologías con kernel.

```
while True:
    "Lectura de la Cámara"
    ret, frame = cap.read()
    if ret==True:
        "Conversión BGR a HSV"
        frameHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        "Máscara de Color para Identificar el Aguacate"
        mask = cv2.inRange(frameHSV, ri, rs)
        "Aplicamos morfologías"
        fmask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
        fmask = cv2.morphologyEx(fmask, cv2.MORPH_CLOSE, kernel)
        fmask = cv2.dilate(fmask, None, iterations = 2)
        "Búsqueda de Contornos con la Máscara de Color"
        contornos, _ = cv2.findContours(fmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        "Ciclo para Identificar Contornos"
```

De esta manera se realiza la detección de color aplicando las morfologías como se observa en la Figura 21 y 22.

Figura 21 Aplicación de la detección de color y área.

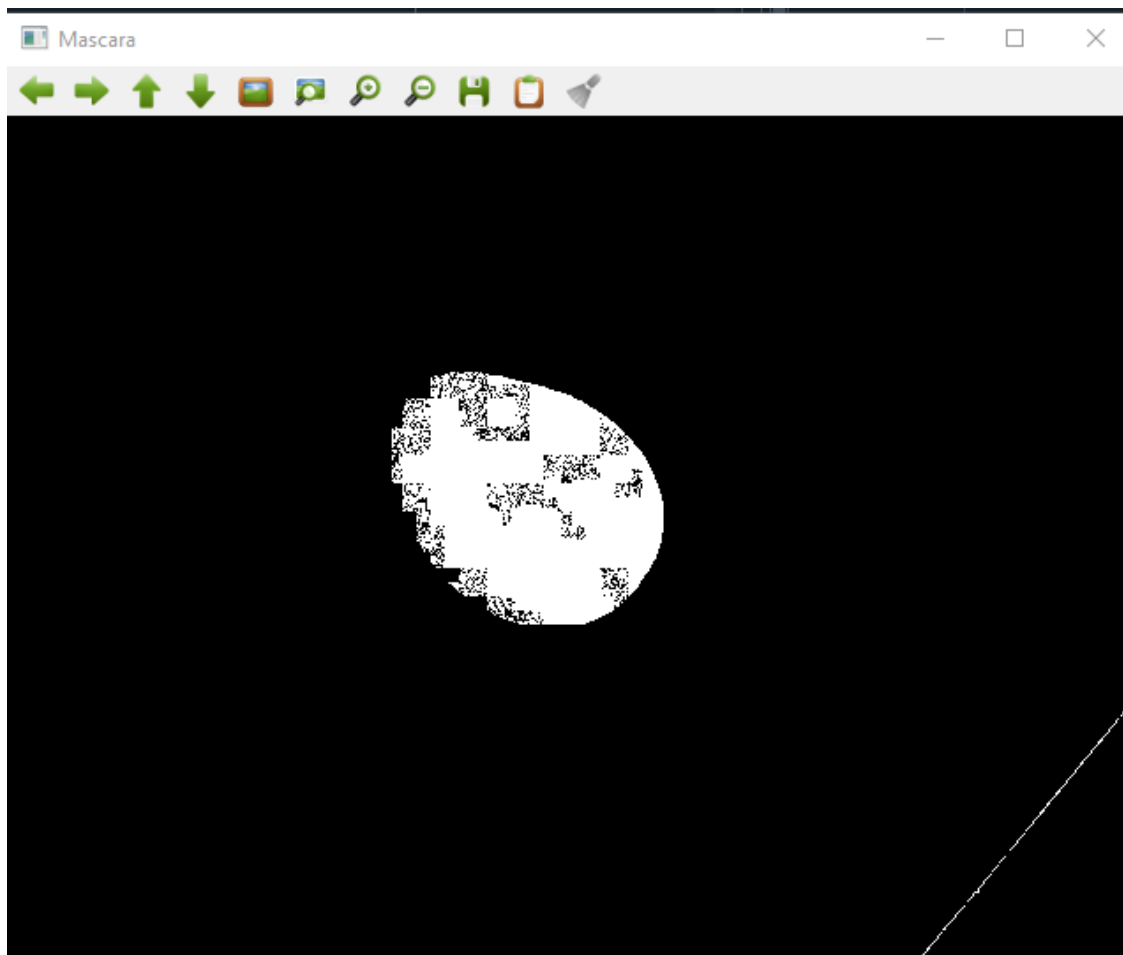
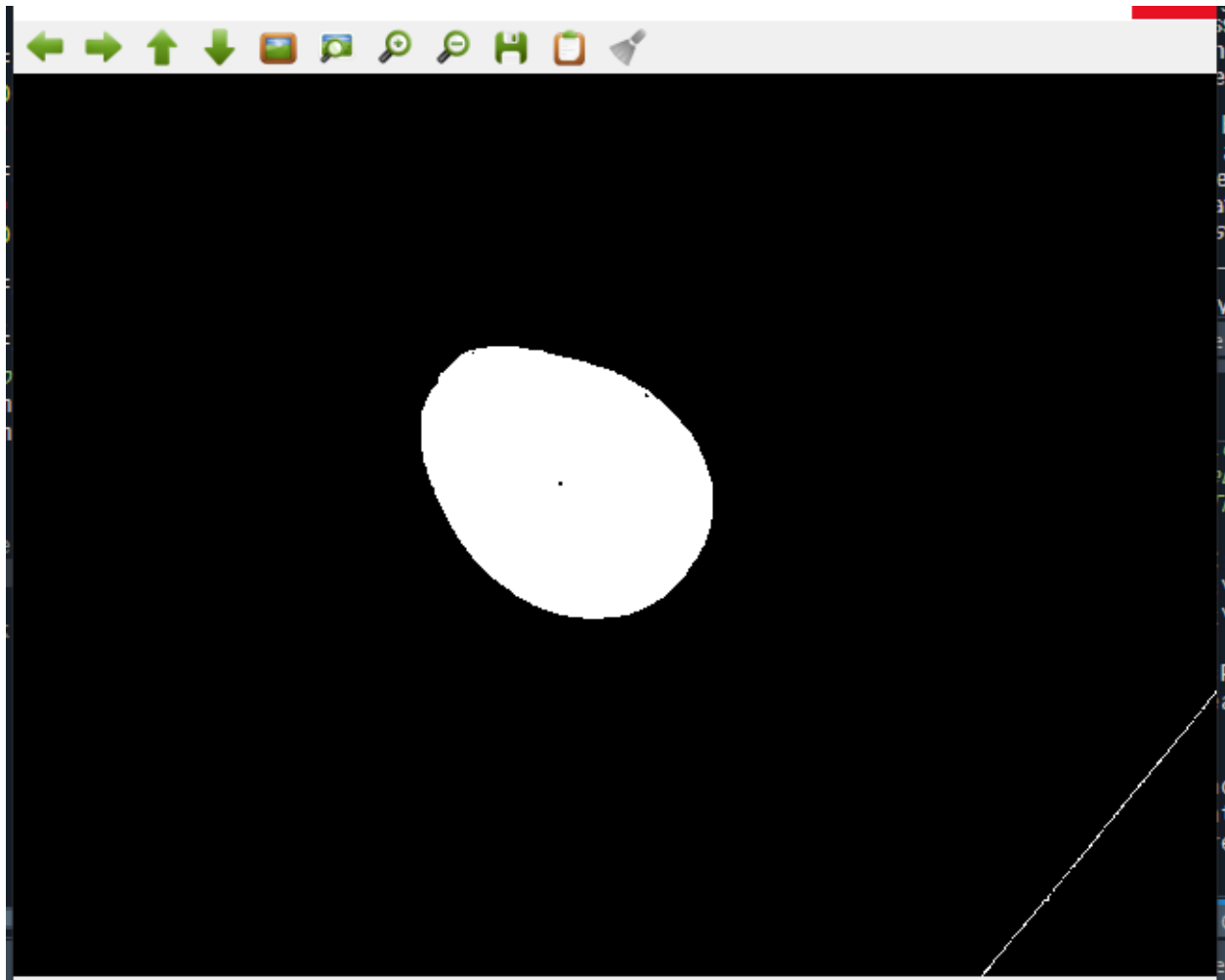


Figura 22 *Aplicación de morfologías con kernel.*



3.7.3. Estimación de Peso

Para la estimación del peso es importante tener en cuenta que el algoritmo genera áreas a partir de la detección de color en magnitud de pixeles, para este mediante pruebas se realiza diferentes cálculos los cuales relacionan un área conocida en cm con el valor en pixeles detectada por el algoritmo. En la Figura 23 se observa en detalle el código.

Figura 23 Códigos de conversión y estimación del peso.

```
for c in contornos:  
    "A Partir de los Contornos Generar un Rectángulo"  
    Rectangulo = cv2.minAreaRect(c)  
    "Extraer Valores del Rectángulo"  
    (x, y), (an, al), angulo = Rectangulo  
    #print(an)  
    #print(al)  
    "Sacar Área de Contornos en Pixeles"  
    areaV = cv2.contourArea(c)  
    "Convertir Área en cm"  
    areaR = ((areaV*10.081)/5465.5)  
    areaR = str("{0:.3f}".format(areaR))  
    "Convertir Altura y Ancho en cm"  
    altura = (al*3.175)/72  
    alturaR = str("{0:.3f}".format(altura))  
    ancho = (an*3.175)/72  
    anchoR = str("{0:.3f}".format(ancho))  
    "Calcular el Volumen del Aguacate"  
    volumen = ((math.pi*altura*pow(ancho,2))/6)*0.7  
    volumenR = str("{0:.3f}".format(volumen))  
    "Calcular el Peso del Aguacate"  
    peso = (volumen*0.630)  
    pesoR = str("{0:.3f}".format(peso))
```

En la Figura 24 se puede observar una comparativa del peso obtenido en una balanza digital en gramos en relación con el valor presentado por el algoritmo.

Figura 24 Comparativa de valores de peso.



Nota: En la ventana de la izquierda se observa la generación del peso estimado y en la venta de la derecha el peso en una balanza digital.

3.7.4. Aplicación de etiquetas

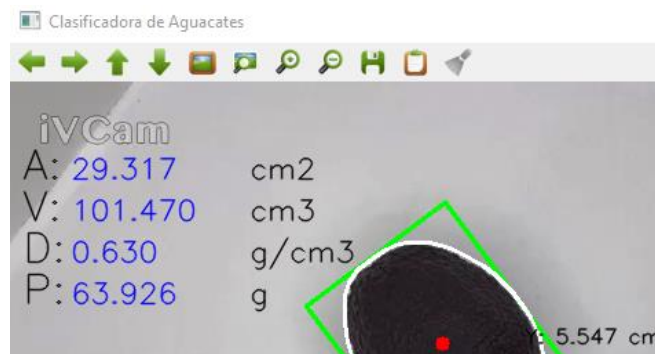
Mediante las líneas de programación de la Figura 25 se generan todas las etiquetas que brindan información.

Figura 25 Código para la generación de etiquetas informativas.

```
if areaV > 500:
    Ncontornos = cv2.convexHull(c)
    "Crear Rectangulo que Contiene al Aguacate"
    rect = cv2.boxPoints(Rectangulo)
    rect = np.int0(rect)
    "Dibujar Rectangulo que Contiene al Aguacate"
    cv2.polylines(frame, [rect], True,(0,255,0),2)
    "Dibujar Centro de Rectangulo que Contiene al Aguacate"
    cv2.circle(frame, (int(x),int(y)),5,(0,0,255),-1)
    "Dibujar los contornos"
    cv2.drawContours(frame, [Ncontornos], -1, (255,255,255), 2)
    "Colocar Texto de Area"
    cv2.putText(frame, 'A:',(10,70), font, 1,(0,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, areaR,(45,70), font, 0.7,(255,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, 'cm2',(175,70), font, 0.7,(0,0,0),1,cv2.LINE_AA)
    "Colocar Texto de Volumen"
    cv2.putText(frame, 'V:',(10,100), font, 1,(0,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, volumenR,(45,100), font, 0.7,(255,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, 'cm3',(175,100), font, 0.7,(0,0,0),1,cv2.LINE_AA)
    "Colocar Texto de Densidad"
    cv2.putText(frame, 'D:',(10,130), font, 1,(0,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, "0.630", (45,130), font, 0.7,(255,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, 'g/cm3',(175,130), font, 0.7,(0,0,0),1,cv2.LINE_AA)
    "Colocar Texto de Peso"
    cv2.putText(frame, 'P:',(10,160), font, 1,(0,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, pesoR,(45,160), font, 0.7,(255,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, 'g',(175,160), font, 0.7,(0,0,0),1,cv2.LINE_AA)
    "Colocar Valores de Altura y Ancho"
    cv2.putText(frame, 'X: {} cm'.format(anchoR),(int(x-40),int(y+100)), font, 0.5,(0,0,0),1,cv2.LINE_AA)
    cv2.putText(frame, 'Y: {} cm'.format(alturaR),(int(x+60),int(y)), font, 0.5,(0,0,0),1,cv2.LINE_AA)
```

Finalmente, en la ventana principal al ejecutarse el algoritmo de clasificación se puede visualizar una ventana interactiva con los gráficos y líneas de dimensionamiento en relación cm y pixeles como se observa en la Figura 26.

Figura 26 Ventana con el algoritmo funcional y presentación de etiquetas.



3.8. Funcionamiento del Algoritmo: Hardware

Para el funcionamiento del hardware se presenta varias procesos y aspectos que se describen a continuación.

3.8.1. Ambiente Controlado

En la clasificación mediante visión artificial contar con un ambiente controlado puede ser un factor crucial al momento de obtener la imagen real. El ambiente controlado construido consta de la estructura de madera, el teléfono celular que es equivalente a una cámara, la iluminación Led, y finalmente el computador que ejecuta el algoritmo. En la Figura 27 se observan las conexiones de los equipos a utilizarse.

Figura 27 Diagrama de conexiones hardware.



3.8.2. Iluminación

En la estructura del ambiente controlado se encuentra un acople donde el aro de luz se coloca de manera exacta fijándose para iluminar el área de trabajo. En la Figura 28 se observa el funcionamiento del aro led, el cual, cuenta con un control de intensidad independiente.

Figura 28 Colocación y funcionamiento del aro de luz led en el ambiente controlado.

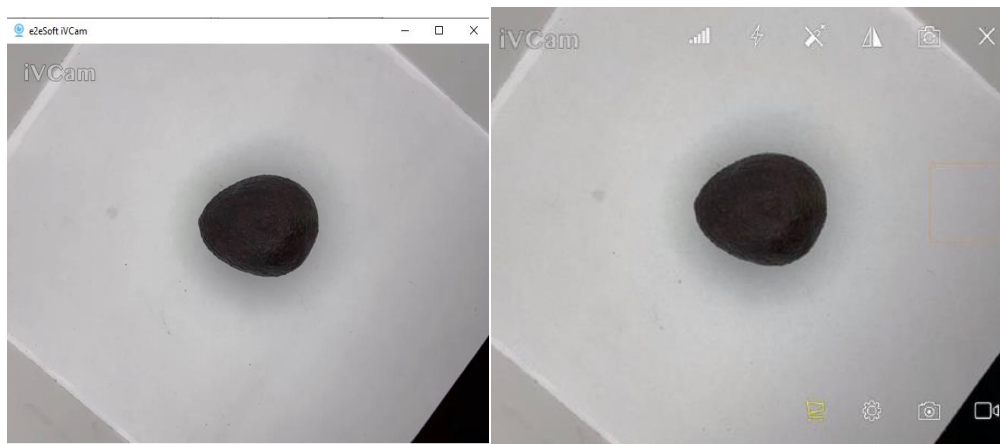


Nota: En las imágenes se observa la colocación y funcionamiento del aro led en la estructura de ambiente controlado.

3.8.3. Periférico de imagen

Para el caso de la cámara la estructura cuenta con una estructura donde el celular se coloca para ser empleado como el ingreso de imagen mediante la aplicación iVCam. Esta conexión se efectúa mediante la red WiFi. En la Figura 29 se observa la interfaz cuando se encuentra enlazada a la aplicación.

Figura 29 Entorno de iVCam conectado con el computador.



Nota: En la ventana de la izquierda esta interfaz de la aplicación de iVCam en la computadora y en la ventana derecha se observa la interfaz de la aplicación de iVCam en el dispositivo móvil, cuando esta generado la conexión.

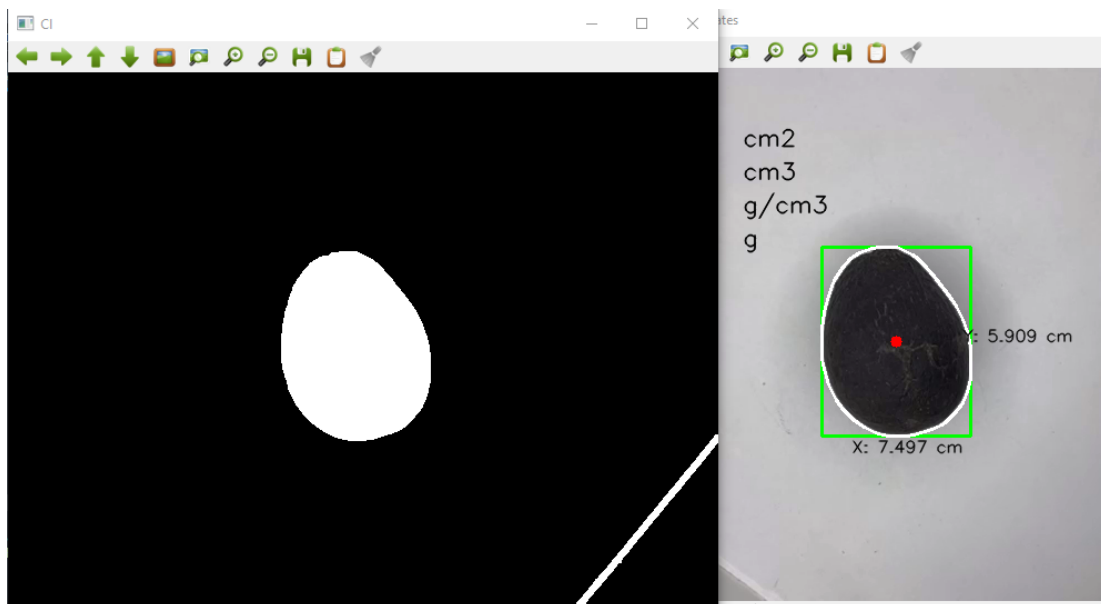
3.9. Pruebas y Calibración

Con el algoritmo funcionando se empieza con la fase de pruebas para verificar que todos los procesos se encuentren funcionando en armonía y observar pequeños errores que fueron obviados y pequeños ajustes o calibraciones que permitan un mejor funcionamiento.

3.9.1. Pruebas

Para las primeras pruebas de funcionamiento como se visualiza en la Figura 30. Se verifico mediante tres pruebas esenciales.

Figura 30 *Funcionamiento del algoritmo de clasificación de aguacates.*

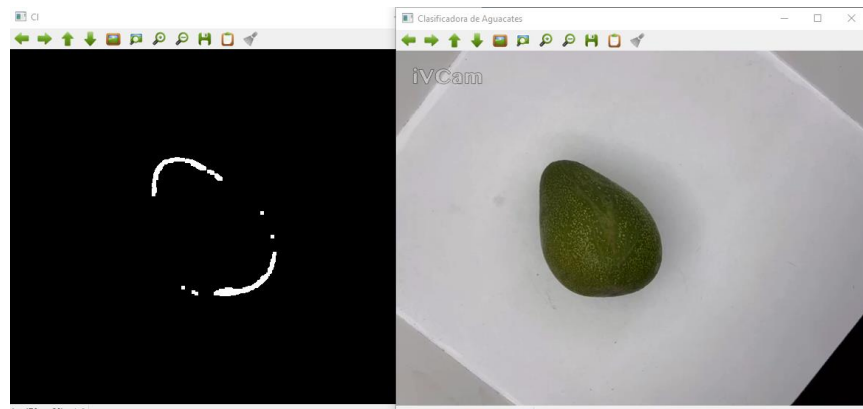


Nota: En la ventana de la izquierda se observa la ventana de detección de color y en la venta de la derecha el funcionamiento de la generación contornos y dimensiones.

3.9.2. Identificación de Aguacates Hass

La primera prueba realizada pone a prueba el funcionamiento de la detección de color para esto se colocan diferentes aguacates de la variante Hass como se observa en la Figura 31.

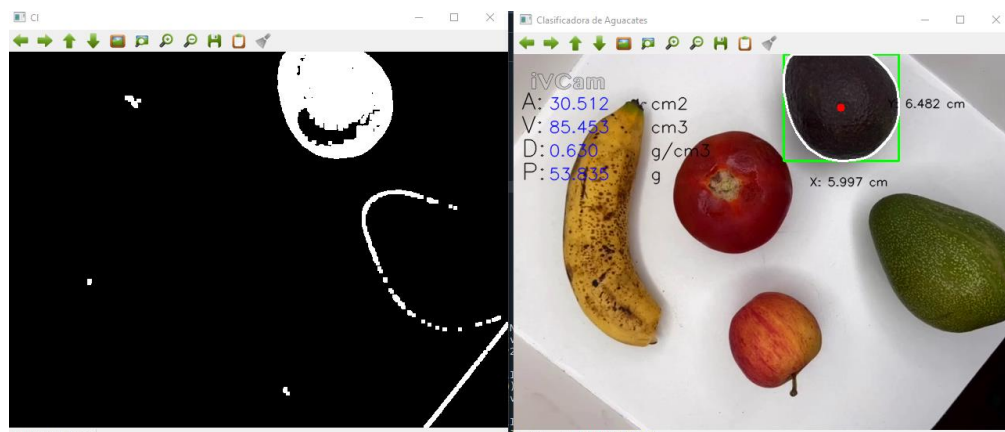
Figura 31 Detección de Aguacates de la Variedad Hass.



Nota: En la ventana de la izquierda se observa la ventana de detección de color con el aguacate tipo Fuerte y en la ventana derecha la ausencia de detección de contornos y presentación de información de clasificación.

A continuación, se realizan pruebas colocando la variedad Fuerte y otro tipo de frutas, Figura 32.

Figura 32 Detección de aguacates y ausencia de detección en otras frutas.



Nota: En la ventana de la izquierda se observa la ventana de detección de color con el aguacate tipo Fuerte, tomate riñón, plátano, manzana y el aguacate tipo Hass y en la ventana derecha la presentación de detección de contornos y presentación de información de clasificación para el aguacate Hass.

En el caso de encontrar un aguacate de variante Hass se visualizan las etiquetas con los valores e información, además de los trazos y figuras que encierran al aguacate.

3.9.3. *Peso Estimado*

En esta prueba se verifica el análisis del área que se detecta a partir del color y el cálculo matemático para generar un valor de peso en gramos. En esta prueba se comparan los valores presentados por el algoritmo con una medición realizada en una balanza digital, Figura 33.

Figura 33 Pesos estimado por el algoritmo y la medición en balanza digital.



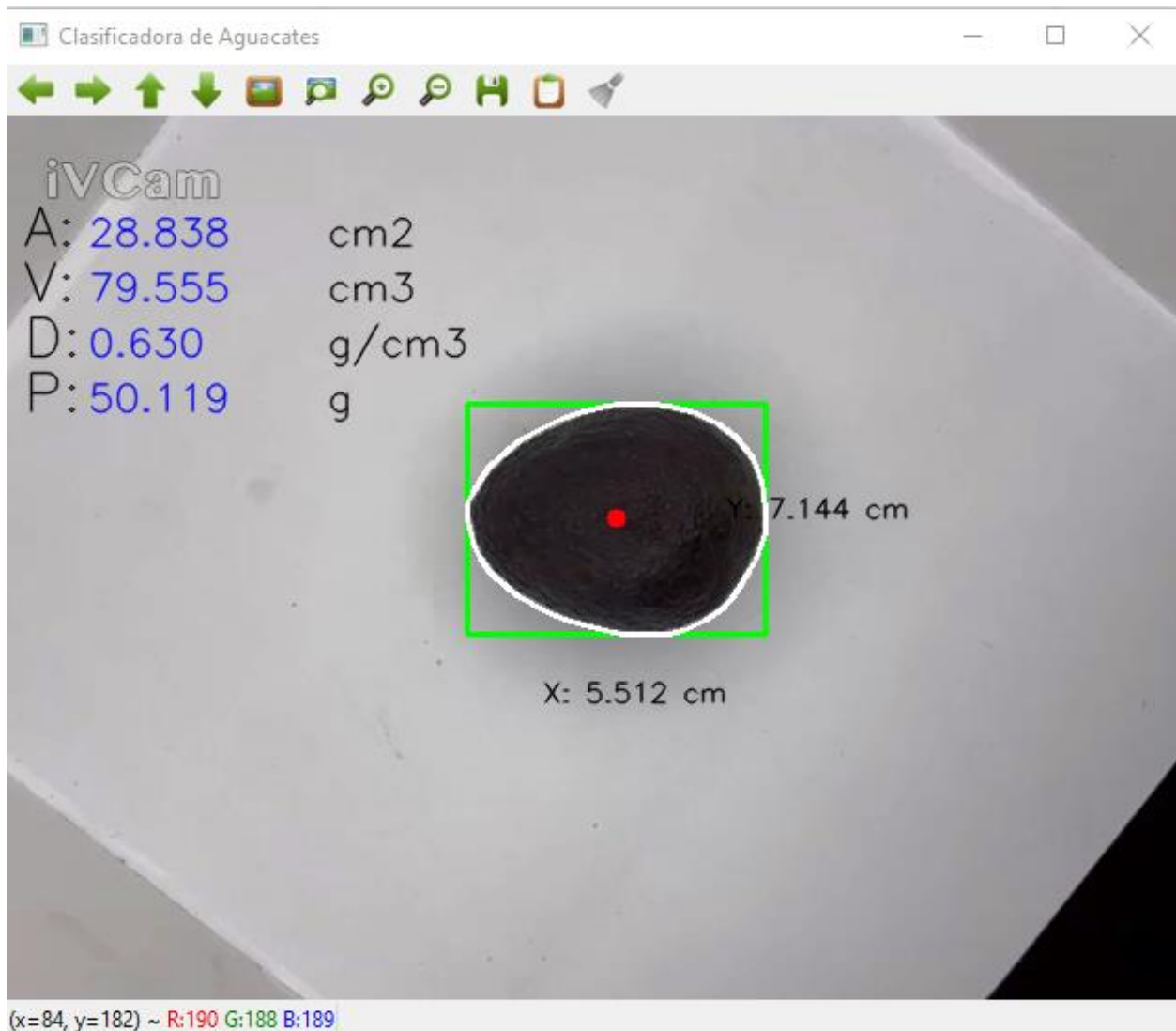
Nota: En la ventana de la izquierda se observa una nueva prueba de la generación del peso estimado y en la ventana de la derecha el peso en una balanza digital.

3.9.4. *Clasificación de Aguacates*

En esta sección es importante recordar la categoría que se coloca a los aguacates según la normativa NTE INEN 1755 2° Edición. En la Tabla 1 previa se presenta el peso en gramos y el grado de clasificación.

En la Figura 34 se observa el algoritmo en funcionamiento.

Figura 34 Clasificación de aguacates con el algoritmo de clasificación.



3.9.5. Calibración

En el aspecto de calibración se pudo evidenciar que se debe realizar un ajuste en los valores de colores en formato HSV utilizando el código independiente de detección de rangos de color cuando se cambie de teléfono celular o en caso de aplicar una cámara web. En caso de sustituir el celular se debe colocar el nuevo dispositivo en el soporte del ambiente controlado y ser ajustado

con un aguacate o varios de prueba. En caso de ser reemplazada por una cámara se recomienda una resolución mínima de 720 pixeles, 2 Megapíxeles y autoenfoque.

3.10. Resultados

De manera consecutiva a las pruebas realizadas anteriormente se presentan los resultados de cada prueba con su análisis.

3.10.1. Identificación de Aguacate Hass.

Para la prueba de identificación del aguacate de variedad Hass se utilizaron 30 frutas donde se registra la coincidencia de la detección como se observa en la Tabla 9. En esta prueba se utilizaron los dos tipos de aguacate y frutas adicionales constituyendo 30 elementos.

Tabla 9 Registro de la identificación de Aguacates de la Variantes Hass.

Nº	TIPO DE AGUACATE	CLASIFICACIÓN POR EL ALGORITMO	COINCIDENCIA
1	HASS	HASS	SI
2	HASS	HASS	SI
3	HASS	HASS	SI
4	HASS	HASS	SI
5	HASS	HASS	SI
6	FUERTE	/	NO
7	FUERTE	/	NO
8	FUERTE	/	NO
9	FUERTE	/	NO
10	FUERTE	/	NO
11	FUERTE	/	NO
12	TOMATE	/	NO
13	HASS	HASS	SI
14	FUERTE	/	NO
15	HASS	HASS	SI
16	FUERTE	/	NO
17	BANANO	/	NO
18	HASS	HASS	SI
19	HASS	HASS	SI
20	TOMATE	/	NO

21	TOMATE	/	NO
22	BANANO	/	NO
23	HASS	HASS	SI
24	FUERTE	/	NO
25	FUERTE	/	NO
26	HASS	HASS	SI
27	FUERTE	/	NO
28	HASS	HASS	SI
29	HASS	HASS	SI
30	HASS	HASS	SI

3.10.2. Análisis de Resultados de Identificación de Aguacate Hass

Finalizada la primera prueba se puede verificar que la detección de color es correcta, en la Tabla 10 se encuentra una cuantificación de esta prueba.

Tabla 10 Eficiencia de la detección de color.

Muestra	30
Coincidencias	30
Error	0
Eficiencia de la detección de color	100.00%

3.10.3. Peso Estimado.

Para la prueba de estimación de peso del aguacate de variedad Hass se utilizaron 30 muestras comparando la medición estimada del algoritmo con la medición de la balanza digital. Esta información se presenta en la Tabla 11.

Tabla 11 Registro de la comparativa de mediciones de peso.

N°	PESO POR ALGORITMO (g)	PESO BALANZA DIGITAL (g)
1	80,25	80,50
2	82,06	82,20

3	90,82	90,50
4	92,63	92,60
5	125,55	125,50
6	95,41	95,40
7	102,09	102,30
8	100,46	100,50
9	102,44	102,40
10	90,47	90,50
11	93,54	93,50
12	93,49	93,50
13	95,64	95,60
14	99,02	99,00
15	99,28	99,30
16	95,28	95,30
17	127,04	127,00
18	125,05	125,00
19	110,48	110,50
20	109,89	109,90
21	89,35	89,40
22	88,30	88,30
23	89,39	89,40
24	127,03	127,00
25	95,32	95,30
26	106,62	106,60
27	107,30	107,30
28	108,44	108,40
29	100,01	99,90
30	99,47	99,40

El siguiente paso es calcular el error porcentual el cual se describe mediante la siguiente fórmula:

$$\text{Error Porcentual} = (\text{Valor Real} - \text{Valor Medido}) / \text{Valor Real} * 100\%$$

De esta manera se obtiene la siguiente Tabla 12 con los errores porcentuales individuales.

Tabla 12 Errores porcentuales de las mediciones.

Nº	PESO POR ALGORITMO	ERROR PORCENTUAL
1	80,25	0,31%

2	82,06	0,17%
3	90,82	0,35%
4	92,63	0,03%
5	125,55	0,04%
6	95,41	0,01%
7	102,09	0,21%
8	100,46	0,04%
9	102,44	0,04%
10	90,47	0,03%
11	93,54	0,04%
12	93,49	0,01%
13	95,64	0,04%
14	99,02	0,02%
15	99,28	0,02%
16	95,28	0,02%
17	127,04	0,03%
18	125,05	0,04%
19	110,48	0,02%
20	109,89	0,01%
21	89,35	0,06%
22	88,30	0,00%
23	89,39	0,01%
24	127,03	0,02%
25	95,32	0,02%
26	106,62	0,02%
27	107,30	0,00%
28	108,44	0,04%
29	100,01	0,11%
30	99,47	0,07%

Adicionalmente, en esta prueba donde se comparan los valores de pesos del algoritmo con la balanza digital, se calculó el coeficiente de correlación de Pearson. A continuación, de manera secuencial se describen los pasos consecutivos que se deben realizar para calcular el coeficiente.

Organización de los datos en dos conjuntos, los valores que corresponden a X y los valores que corresponden a Y. en este proyecto X son los valores medidos por el algoritmo y Y los valores medidos por la balanza digital

Cálculo de la media, se realiza la suma de todas las mediciones (30) y se divide para el total de mediciones. Esto se realiza para cada conjunto de datos.

Cálculo de la desviación estándar, mediante este cálculo se determina la desviación de los pesos estimados y los pesos medidos.

Cálculo de la covarianza, en este cálculo se conoce cuánto varían las mediciones con respecto a sus medias aritméticas.

Cálculo del coeficiente de correlación Pearson, se utiliza la covarianza y las desviaciones estándar para obtener el valor de -1, 0, 1. Este coeficiente se calcula mediante la siguiente ecuación:

$$\text{Coeficiente Pearson} = \text{Covarianza} / (\text{Desviación Estándar Algoritmo} * \text{Desviación Estándar Balanza})$$

Utilizando la herramienta de [26], una calculadora para el coeficiente de correlación Pearson se obtiene el siguiente valor, Figura 35 :

Figura 35 Calculadora de coeficiente de Pearson.

Los datos deben separarse por un espacio e introducirse usando el punto como separador decimal.

- Variable aleatoria X:
80.25 82.06 90.82 92.63 125.55 95.41 102.09 100.46 102.44 90.47 93.54
93.49 95.64 99.02 99.28 95.28 127.04 125.05 110.48 109.89 89.35 88.30
89.39 127.03 95.32 106.62 107.30 108.44 100.01 99.47
- Variable aleatoria Y:
80.50 82.20 90.50 92.60 125.50 95.40 102.30 100.50 102.40 90.50 93.50
93.50 95.60 99.00 99.30 95.30 127.00 125.00 110.50 109.90 89.40 88.30
89.40 127.00 95.30 106.60 107.30 108.40 99.90 99.40

Calcular

→ Coeficiente de correlación de Pearson = 1

Con este valor obtenido por la calculadora online, valor de 1. Sin embargo, para ser más exacto utilizando las funciones y propiedades de Excel se obtiene el valor de 0,966 lo cual permite deducir que existe una alta correlación entre los valores de peso estimados del algoritmo y los pesos medidos por la balanza digital, permitiendo enunciar que el algoritmo presenta una alta exactitud en sus estimaciones de peso.

Adicionalmente, generando el coeficiente de determinación (R^2), $R^2 = 0,966 * 0,996 = 0,9331$, significa que el 93,33% de la mediciones son semejantes (variabilidad) entre los pesos de los aguacates estimados por el algoritmo y los pesos estimados por la balanza.

3.10.4. Análisis de Resultados de Peso Estimado

Finalizada la segunda prueba se realiza un promedio del error porcentual para ver su eficiencia el cual debe ser mayor a 10% para concluir como funcional el algoritmo [27].

Promedio = Sumatoria de Datos a Promediar / Cantidad de Datos Sumados

Promedio de errores con respecto a la balanza digital: 0,06%

3.10.5. Clasificación de Aguacates.

Para la prueba de clasificación del aguacate de variedad Hass se utilizaron 30 muestras obteniendo la siguiente información, Tabla 14. Es importante mencionar que en base a la masa del aguacate Hass de tiene la siguiente categorización, Tabla 13.

Tabla 13 Categorías del aguacate Hass según su masa.

PESO (g)	CAETGORÍA
>125	A+
120 < m < 125	A
100 < m < 120	B
80 < m < 100	C
< 80	C-

Tabla 14 *Categorías del aguacate Hass según su masa.*

N°	CLASIFICACIÓN POR ALGORITMO	CLASIFICACIÓN POR PRODUCTOR	COINCIDENCIA
1	C	C	SI
2	C	C	SI
3	C	C	SI
4	C	C	SI
5	A+	A+	SI
6	C	C	SI
7	B	B	SI
8	B	B	SI
9	B	B	SI
10	C	C	SI
11	C	C	SI
12	C	C	SI
13	C	C	SI
14	C	C	SI
15	C	C	SI
16	C	C	SI
17	A+	A+	SI
18	A+	A+	SI
19	B	B	SI
20	B	B	SI
21	C	C	SI
22	C	C	SI
23	C	C	SI
24	A+	A+	SI
25	C	C	SI
26	B	B	SI
27	B	B	SI
28	B	B	SI
29	C	B	NO
30	C	C	SI

3.10.6. Análisis de Resultados de Clasificación de Aguacates

Finalizada la tercera prueba se puede verificar que clasificación es correcta, en la Tabla 15 se encuentra una cuantificación de esta prueba.

Tabla 15 *Eficiencia de la clasificación de aguacates por su masa.*

Muestra	30
Coincidencias	29
Error	1
Eficiencia de la detección de color	96.67%

Trabajos anteriores y de los cuales se han servido de punto de partida en los antecedentes trabajan la aplicación de sistemas electrónicos, de control o de programación que permita introducirse en la agricultura de precisión para buscar una optimización de sus recursos y con ello generar un menor impacto en el ambiente y mantener o aumentar su utilidad en el aspecto comercial.

En el aspecto electrónico y mecánico se propone sistemas que constituyen maquinaria de unos tamaños y costos considerables donde su funcionalidad permite generar una clasificación automática o semiautomática de la fruta, en este caso el aguacate. La sensórica, los actuadores permiten dar funcionamiento a estos sistemas.

Este proyecto que se desarrolló basado en la programación mediante visión artificial la cual busco presentarse como una alternativa para la clasificación de aguacates. Después de realizar las tres pruebas que evalúa el funcionamiento y la capacidad del algoritmo es importante recalcar sus resultados ya que estos sistemas no necesitan de un gran tamaño o equipo para su funcionamiento, además, su versatilidad permitiría que este sistema se adapte a otro tipo de frutas o verduras.

Un error porcentual promedio de 0,06% y una eficiencia en la detección y análisis de peso del 96,67% muestra esta alternativa factible para ayudar a optimizar los procesos de selección en el caso de pequeños y medianos productores sin repercutir como una inversión de gran costo.

CONCLUSIONES

El análisis de información realizado en este proyecto acerca del aguacate y sus dos variedades, Fuerte y Hass, reveló que las características físicas son los indicadores clave que el algoritmo debe identificar y poder clasificarlos. Estas características incluyen el color, tamaño, largo y forma.

Después de examinar las características del aguacate tipo Hass, se determinaron los aspectos más relevantes para su clasificación. Específicamente, dos tonos de color verdes permitieron generar el área identificada y generar una forma casi esférica para aplicar la matemática y generar una relación cm/píxel estimando un peso aproximado siendo estos los fundamentos sobre los cuales se diseñó el algoritmo propuesto, el cual emplea técnicas de visión artificial.

El algoritmo propuesto se estableció como una herramienta que mejora el proceso de clasificación de aguacate Hass en el sector de producción agrícola homogeneizando los tamaños, además, demostró del 96,66% de eficiencia en su identificación resultando ser ideal como punto de partida para la implementación de este y otros sistemas en la optimización de la producción orientada hacia la agricultura de precisión.

Se realizaron diversas pruebas de validación del algoritmo con la colaboración de productores de aguacates, basándose en los estándares de la normativa NTE INEN 1755 2da Edición, abarcando la identificación de diferentes tipos, tamaños y formas. Estas pruebas demostraron el funcionamiento total del algoritmo para el aguacate tipo Hass, lo que sugiere la posibilidad de adaptarlo para su aplicación en su otro tipo, Fuerte.

RECOMENDACIONES

Para obtener resultados precisos y consistentes, es beneficioso contar con un entorno controlado o un medio que permita aplicar la visión artificial de manera uniforme a un gran número de muestras. Esto facilita la recolección de información en las mismas condiciones y la generación de una base de datos para satisfacer diferentes necesidades.

Es recomendable vincular la aplicación de un algoritmo de visión artificial con sistemas de actuadores u otros sistemas adicionales que utilicen la información obtenida a través de la identificación de la madurez u otras características. De esta manera, se puede aprovechar al máximo la información recopilada y utilizarla para la toma de decisiones o la implementación de acciones específicas.

El uso de hardware de alto rendimiento, como una cámara de excelente calidad y alta resolución, potencia los resultados al aplicar cualquier algoritmo de visión artificial.

REFERENCIAS

- [1] M. de A. y Ganadería, “Aguacate será producto estratégico para desarrollar el agro,” 2020. <https://www.agricultura.gob.ec/aguacate-sera-producto-estrategico-para-desarrollar-el-agro/>
- [2] G. Macas, “Estudio de las características de calidad pre y poscosecha en dos variedades de aguacate (*Persea americana* Mili) provenientes de dos localidades de la provincia de Pichincha,” 2013. Accessed: Jun. 09, 2023. [Online]. Available: <https://repositorio.iniap.gob.ec/bitstream/41000/986/1/iniapscP.M113e2012.pdf>
- [3] T. Menéndez, “Productores aumentan cultivos de aguacate para entrar a Estados Unidos,” 2023. [Online]. Available: [https://www.primicias.ec/noticias/economia/productores-aumentan-cultivos-ingreso-aguacate-ecuatoriano-estados-unidos/#:~:text=Ecuador exportó 600 toneladas de,%2C54%25 a Hong Kong.2023](https://www.primicias.ec/noticias/economia/productores-aumentan-cultivos-ingreso-aguacate-ecuatoriano-estados-unidos/#:~:text=Ecuador%20export%C3%B3%20600%20toneladas%20de%20,2C54%25%20a%20Hong%20Kong.2023)
- [4] INEN, “Frutas frescas. Aguacate. Requisitos NTE INEN 1755,” *Inst. Ecuatoriano Norm.*, no. Frutas frescas. Aguacate. Requisitos, pp. 1–15, 2015, [Online]. Available: <https://www.normalizacion.gob.ec/buzon/normas/nte-inen-1755-2.pdf>
- [5] L. Burbano, “Máquina para clasificar aguacates por su peso.,” 2015. Accessed: Jun. 09, 2023. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/4604>
- [6] D. Figueroa, “Diseño e implementación de una herramienta computacional para la identificación del estado de madurez de la granadilla mediante técnicas de visión artificial sobre un ordenador de placa reducida,” 2016, [Online]. Available: <https://repositorio.ucundinamarca.edu.co/handle/20.500.12558/169>

- [7] S. De la Torre, “SISTEMA DE CLASIFICACIÓN DE HUEVOS MEDIANTE UN ALGORITMO DE VISIÓN ARTIFICIAL,” 2022.
- [8] J. León, “Manual del cultivo del aguacate (*Persea americana*) para los Valles interandinos del Ecuador,” 1999. [Online]. Available: [https://repositorio.iniap.gob.ec/handle/41000/51#:~:text=Resumen %3A,Tungurahua las de mayor extensión.](https://repositorio.iniap.gob.ec/handle/41000/51#:~:text=Resumen%3A,Tungurahua%20las%20de%20mayor%20extensi3n)
- [9] Reyes Gutiérrez, “Aguacate,” 2023. <https://www.interempresas.net/Distribucion-Hortofruticola/FeriaVirtual/Producto-Aguacate-Fuerte-75326.html>
- [10] CALAMEO, *Manual Poscosecha Del Aguacat.* 2023. [Online]. Available: <https://www.calameo.com/books/0054748321fa3f0ead0ea>
- [11] UNISORTING, “Máquinas clasificadoras, seleccionadoras y sistemas completos para empaque y procesamiento de paltas (aguacates).,” 2023. <https://www.unisorting.com/es/productos/paltas-calibradoras-empaque-procesamiento/>
- [12] I. Pérez, Y. Díaz, and A. Becerra, “El lenguaje de programación Python,” 2022. [Online]. Available: <http://www.linuxjournal.com/article/2959>
- [13] D. Ramírez, G. Pulido, B. Gerardino, J. Cruz, E. Estupiñán, and S. Cancino, “Adquisición y Reconocimiento de Imágenes por Medio de Técnicas de Visión e Inteligencia Artificial,” 2009.
- [14] Aprende con Alf, “La librería Numpy,” 2022. <https://aprendeconalf.es/docencia/python/manual/numpy/> (accessed Oct. 25, 2022)
- [15] V. Arévalo, J. Gonzáles, and G. Ambrosio, “LA LIBRERÍA DE VISIÓN ARTIFICIAL

OPENCV APLICACIÓN A LA DOCENCIA E INVESTIGACIÓN,” 2018.

- [16] C. Rondón, A. Castro, and B. Medina, “Caracterización para la ubicación en la captura de video aplicado a técnicas de visión artificial en la detección de personas,” 2020.
- [17] Z. Effendi, R. Ramli, and J. A. Ghani, “A back propagation neural networks for grading *Jatropha curcas* fruits maturity,” *Am. J. Appl. Sci.*, vol. 7, no. 3, pp. 390–394, 2010, doi: 10.3844/AJASSP.2010.390.394.
- [18] V. Alvear-Puertas, P. Rosero-Montalvo, D. Peluffo-Ordóñez, and J. Pijal-Rojas, “Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura,” *Enfoque UTE*, vol. 8, no. 1, pp. 244–256, Feb. 2017, doi: 10.29019/ENFOQUEUTE.V8N1.121.
- [19] D. Farrugia, “Introducing Python’s Magic Methods | by David Farrugia | May, 2023 | Towards Data Science,” 2022. <https://towardsdatascience.com/introducing-pythons-magic-methods-f443ed913703> (accessed Jun. 29, 2023).
- [20] A. Marzal, V. Isabel, G. Luengo, and P. G. Sevilla, *Introducción a la programación con Python 3*. 2022. doi: 10.6035/Sapientia93.
- [21] Ministerio de Educación del Gobierno de España, *Aplicación práctica de la visión artificial en el control de procesos industriales*. 2012. [Online]. Available: <http://www.adimenlehiakorra.eus/documents/29934/43025/Aplicación+práctica+de+la+visión+artificial+en+el+control+de+procesos+industriales.pdf/92a1a455-ffe2-4290-be95-f9ba33c02241>
- [22] Clearview, “Vision Articial: Una guía introductoria,” 2018. <https://www.clearview->

imaging.com/es/vision-artificial-y-aplicaciones-automatizacion-industrial (accessed Jun. 29, 2023).

[23] R. Hernández, C. Fernández, and P. Baptista, *Metodología de la Investigación*. 1997.

[24] S. Laoyan, “Qué es la metodología waterfall y cuándo utilizarla,” 2022.

[25] P. Santamaria, “Cómo elegir la bombilla LED correcta para cada necesidad,” 2021.

<https://www.xatakahome.com/iluminacion-y-energia/como-elegir-bombilla-led-correcta-para-hogar#:~:text=Para conocer los lúmenes que,potencia lumínica de 840 lm.>

[26] P. y Estadística, “Coeficiente de Correlación de Pearson,” 2022.

<https://www.probabilidadyestadistica.net/coeficiente-de-correlacion-de-pearson/#calculadora-del-coeficiente-de-correlacion-de-pearson>

[27] M. Santo and G. Lecumberry, *El proceso de medición: Análisis y comunicación de datos experimentales*. 2005. [Online]. Available: <https://www.xatakahome.com/iluminacion-y-energia/como-elegir-bombilla-led-correcta-para-hogar#:~:text=Para conocer los lúmenes que,potencia lumínica de 840 lm.>

ANEXOS

ANEXO 1 Código en Detección de Rangos en HSV

```
import numpy as np
import cv2

def nada(X):
    pass

cv2.namedWindow('Parametros')
cv2.createTrackbar('Tonalidad Minima', 'Parametros', 0, 179, nada)
cv2.createTrackbar('Tonalidad Maxima', 'Parametros', 0, 179, nada) #H
cv2.createTrackbar('Pureza Minima', 'Parametros', 0, 255, nada)
cv2.createTrackbar('Pureza Maxima', 'Parametros', 0, 255, nada) #S
cv2.createTrackbar('Luminosidad Minima', 'Parametros', 0, 255, nada)
cv2.createTrackbar('Luminosidad Maxima', 'Parametros', 0, 255, nada) #V
cv2.createTrackbar('Kernel X', 'Parametros', 1, 30, nada)
cv2.createTrackbar('Kernel Y', 'Parametros', 1, 30, nada) #Filtro Kernel

#Crear el video
cap = cv2.VideoCapture(0)

while(1):
    ret,frame = cap.read()

    if ret:
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
        Tmin = cv2.getTrackbarPos('Tonalidad Minima','Parametros')
        Tmax = cv2.getTrackbarPos('Tonalidad Maxima','Parametros')
        Pmin = cv2.getTrackbarPos('Pureza Minima','Parametros')
        Pmax = cv2.getTrackbarPos('Pureza Maxima','Parametros')
        Lmin = cv2.getTrackbarPos('Luminosidad Minima','Parametros')
```



```

Lmax = cv2.getTrackbarPos('Luminosidad Maxima','Parametros')

color_oscuro = np.array([Tmin, Pmin, Lmin])
color_brilla = np.array([Tmax, Pmax, Lmax])

mascara = cv2.inRange(hsv, color_oscuro, color_brilla)

kernelx = cv2.getTrackbarPos('Kernel X', 'Parametros')
kernely = cv2.getTrackbarPos('Kernel Y', 'Parametros')

kernel = np.ones((kernelx, kernely), np.uint8)
mascara = cv2.morphologyEx(mascara, cv2.MORPH_CLOSE, kernel)
mascara = cv2.morphologyEx(mascara, cv2.MORPH_OPEN, kernel)

contornos, _ = cv2.findContours(mascara, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(frame, contornos, -1, (0, 0, 0), 2)

cv2.imshow("Camara", frame)
cv2.imshow("Mascara", mascara)

if cv2.waitKey(1) & 0xFF == ord('s'):
    break
cap.release()
cv2.destroyAllWindows()

```

ANEXO 2 Código en Lenguaje Python

```
"Librería Opencv"
import cv2
"Librería Numpy"
import numpy as np
"Librería Matemática"
import math
"Lectura Cámara"
cap = cv2.VideoCapture(0)

#cap.set(10,160)
#cap.set(3,1280)
#cap.set(4,720)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))

"Límite Inferior de Color"
ri = np.array([11, 46, 0],np.uint8)
"Límite Superior de Color"
rs = np.array([44, 232, 120],np.uint8)
"Tipo Fuente Texto"
font = cv2.FONT_HERSHEY_SIMPLEX #Tipo de Letra
"Inicio del Ciclo"
while True:
    "Lectura de la Cámara"
    ret,frame = cap.read()
    if ret==True:
        "Conversión BGR a HSV"
        frameHSV = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
        "Máscara de Color para Identificar el Aguacate"
        mask = cv2.inRange(frameHSV,ri,rs)
        "Aplicamos morfologías"
```

```

fmask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
fmask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
fmask = cv2.dilate(fmask, None, iterations = 2)
"Búsqueda de Contornos con la Máscara de Color"
contornos,_ = cv2.findContours(fmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
"Ciclo para Identificar Contornos"
for c in contornos:
    "A Partir de los Contornos Generar un Rectángulo"
    Rectangulo = cv2.minAreaRect(c)
    "Extraer Valores del Rectángulo"
    (x, y), (an, al), angulo = Rectangulo
    #print(an)
    #print(al)
    "Sacar Área de Contornos en Píxeles"
    areaV = cv2.contourArea(c)
    "Convertir Área en cm"
    areaR = ((areaV*10.081)/5465.5)
    areaR = str("{0:.3f}".format(areaR))
    "Convertir Altura y Ancho en cm"
    altura = (al*3.175)/72
    alturaR = str("{0:.3f}".format(altura))
    ancho = (an*3.175)/72
    anchoR = str("{0:.3f}".format(ancho))
    "Calcular el Volumen del Aguacate"
    volumen = ((math.pi*altura*pow(ancho,2))/6)*0.7
    volumenR = str("{0:.3f}".format(volumen))
    "Calcular el Peso del Aguacate"
    peso = (volumen*0.630)
    pesoR = str("{0:.3f}".format(peso))
    if areaV > 500:
        Ncontornos = cv2.convexHull(c)
        "Crear Rectangulo que Contiene al Aguacate"
        rect = cv2.boxPoints(Rectangulo)

```

```

rect = np.int0(rect)
"Dibujar Rectangulo que Contiene al Aguacate"
cv2.polylines(frame, [rect], True,(0,255,0),2)
"Dibujar Centro de Rectangulo que Contiene al Aguacate"
cv2.circle(frame, (int(x),int(y)),5,(0,0,255),-1)
"Dibujar los contornos"
cv2.drawContours(frame, [Ncontornos], -1, (255,255,255), 2)
"Colocar Texto de Área"
cv2.putText(frame,'A:',(10,70), font, 1,(0,0,0),1,cv2.LINE_AA)
cv2.putText(frame,areaR,(45,70), font, 0.7,(255,0,0),1,cv2.LINE_AA)
cv2.putText(frame,'cm2',(175,70), font, 0.7,(0,0,0),1,cv2.LINE_AA)
"Colocar Texto de Volumen"
cv2.putText(frame,'V:',(10,100), font, 1,(0,0,0),1,cv2.LINE_AA)
cv2.putText(frame, volumenR,(45,100), font, 0.7,(255,0,0),1,cv2.LINE_AA)
cv2.putText(frame,'cm3',(175,100), font, 0.7,(0,0,0),1,cv2.LINE_AA)
"Colocar Texto de Densidad"
cv2.putText(frame,'D:',(10,130), font, 1,(0,0,0),1,cv2.LINE_AA)
cv2.putText(frame,"0.630",(45,130), font, 0.7,(255,0,0),1,cv2.LINE_AA)
cv2.putText(frame,'g/cm3',(175,130), font, 0.7,(0,0,0),1,cv2.LINE_AA)
"Colocar Texto de Peso"
cv2.putText(frame,'P:',(10,160), font, 1,(0,0,0),1,cv2.LINE_AA)
cv2.putText(frame, pesoR,(45,160), font, 0.7,(255,0,0),1,cv2.LINE_AA)
cv2.putText(frame,'g',(175,160), font, 0.7,(0,0,0),1,cv2.LINE_AA)
"Colocar Valores de Altura y Ancho"
cv2.putText(frame, 'X: {} cm'.format(anchoR),(int(x-40),int(y+100)),font, 0.5,(0,0,0), 1,cv2.LINE_AA)
cv2.putText(frame, 'Y: {} cm'.format(alturaR),(int(x+60),int(y)),font, 0.5,(0,0,0), 1,cv2.LINE_AA)
#printn(AreaV)

"Mostrar"
#cv2.imshow('Clasificadora de Aguacates',frame)
#cv2.imshow('Clasificadora de Aguacates',mask)
cv2.imshow('Clasificadora de Aguacates',fmask)
"Tecla de salida"

```

```
if cv2.waitKey(1) & 0xFF == ord('s'):
    break
"Finalizar cámara"
cap.release()
"Salida"
cv2.destroyAllWindows()
```