UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE ELECTRÓNICA Y REDES DE COMUNICACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

DISEÑO DE UN SISTEMA EMBEBIDO DE CLASIFICACIÓN DEL GRANO DE LUPINUS MUTABILIS SWEET (CHOCHO) MEDIANTE EL USO DE VISIÓN ARTIFICIAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO EN INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTOR: MARIO DAVID FARINANGO TÚQUERRES

DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA

ASESOR: MSC. LUIS EDILBERTO SUÁREZ ZAMBRANO

Ibarra – Ecuador

2025

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			
CÉDULA DE IDENTIDAD:	1003889308		
APELLIDOS Y NOMBRES:	Farinango Túquerres Mario David		
DIRECCIÓN:	La Esperanza- Comunidad de El Abra		
EMAIL:	mdfarinangot@utn.edu.ec/davidfarinango14@gmail.com		
TELÉFONO FIJO:		**TELÉFONO MÓVIL:**	0969281548

DATOS DE LA OBRA	
TÍTULO:	DISEÑO DE UN SISTEMA EMBEBIDO DE CLASIFICACIÓN DEL GRANO DE LUPINUS MUTABILIS SWEET (CHOCHO) MEDIANTE EL USO DE VISIÓN ARTIFICIAL
AUTOR (ES):	Farinango Túquerres Mario David
FECHA: DD/MM/AAAA	27/01/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	■ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniería en Electrónica y Redes de Comunicación
ASESOR /DIRECTOR:	MSc. Edgar Alberto Maya Olalla MSc. Luis Edilberto Suárez Zambrano

2. CONSTANCIAS

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

Ibarra, a los 21 días del mes de Julio de 2025

EL AUTOR:

.....
Mario David Farinango Túquerres

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

CERTIFICACIÓN:

MAGISTER EDGAR MAYA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que el presente trabajo de Titulación: DISEÑO DE UN SISTEMA EMBEBIDO DE CLASIFICACIÓN DEL GRANO DE LUPINUS MUTABILIS SWEET (CHOCHO) MEDIANTE EL USO DE VISIÓN ARTIFICIAL, ha sido desarrollado por el señor: Farinango Túquerres Mario David bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Edgar Alberto Maya Olalla

DIRECTOR

DEDICATORIA

Este trabajo de titulación está dedicado, con todo mi cariño y gratitud, a mi madre Rosa Matilde y a mi padre José Cesar, por su apoyo incondicional durante toda mi etapa universitaria. Su motivación, esfuerzo y confianza en mí han sido fundamentales para no rendirme y alcanzar este sueño anhelado.

A mi esposa Barbarita y a mi hijo Julián, por ser mi fuente de inspiración y motivación para seguir superándome cada día. Su amor y apoyo constante me han demostrado que los logros se alcanzan con perseverancia, dedicación y esfuerzo

A mis hermanos, por brindarme palabras de aliento y apoyo en los momentos más desafiantes. Sin su compañía y motivación, este proceso habría sido mucho más difícil de sobrellevar.

Finalmente, agradezco a todas las personas que, de manera directa o indirecta, contribuyeron a la realización de este proyecto. Este logro es el resultado de un esfuerzo colectivo, y estoy profundamente agradecido por cada aporte recibido.

Farinango Túquerres Mario David

AGRADECIMIENTO

Quiero expresar mis agradecimientos,

En primer lugar, deseo expresar mi más profundo agradecimiento a Dios y a la Santísima Virgen del Rosario, por haberme brindado salud, vida, paciencia y sabiduría durante todo este proceso. Su guía y bendición han sido fundamentales para culminar esta etapa tan importante en mi carrera profesional.

A mis padres, Rosa y José, por su constante apoyo emocional y motivación durante todo este tiempo de estudio. Su confianza incondicional ha sido un pilar fundamental para alcanzar este logro.

De igual forma, expreso mi más profundo agradecimiento al MSc. Edgar Maya, mi tutor de tesis, y al MSc. Luis Suárez, mi asesor de tesis, por su invaluable guía, dedicación y apoyo durante todo el proceso de investigación. Su gran experiencia, conocimientos impartidos día tras día para el desarrollo y culminación de este trabajo de titulación fue fundamental.

A mis amigos, compañeros de clase, por acompañarme en los momentos más desafiantes y celebrar cada avance. Su compañía hizo este camino más llevadero y significativo para el logro anhelado.

Farinango Túquerres Mario David

RESUMEN

En el Presente trabajo de titulación se describe el desarrollado un sistema embebido para la clasificación del grano de chocho a través de visión artificial, con el propósito de mejorar la selección del grano de chocho postcosecha y de esta manera optimizar el proceso de selección del grano.

Este presente proyecto propone el diseño de un sistema embebido de bajo costo para la clasificación del grano de Lupinus Mutabilis Sweet (chocho), basado en técnicas de visión artificial y aprendizaje profundo. El sistema integra una cámara de alta resolución, una placa embebida (Raspberry Pi 5 y un modelo de red neuronal convolucional (CNN) entrenado para identificar características morfológicas y defectos en los granos de chochos.

Para la determinación del dataset se estableció un numero promedio de 1000 muestras entre diferentes clases de grano como son: grano bueno, grano malo, grano partido y otro tipo de grano (maíz o haba), la respectiva etiquetación de las clases se lo realizó utilizando plataformas libres en este caso se utilizó Robo Flow en el proceso de etiquetado como en el proceso de entrenamiento del dataset. Como siguiente proceso se realizó el entrenamiento de la arquitectura de redes neuronales por lo cual se seleccionó YOLO en su última versión que es Yolov8 Los resultados experimentales demostraron una precisión del 73.3% en la clasificación, validando la eficacia del sistema para optimizar procesos agrícolas. Este trabajo contribuye a la automatización de la agricultura en pequeña escala, reduciendo costos y mejorando la calidad del producto final.

Palabras claves: Deep Learning, Visión artificial, Raspberry, Lupinus Mutabilis Sweet, agricultura de precisión, ultralytics, yolov8.

ABSTACT

In this thesis project, the development of an embedded system for the classification of chocho grains through computer vision is described. The purpose of this system is to improve the post-harvest selection of chocho grains, thereby optimizing the grain selection process.

This project proposes the design of a low-cost embedded system for the classification of *Lupinus mutabilis* Sweet (chocho) grains, based on computer vision and deep learning techniques. The system integrates a high-resolution camera, an embedded board (Raspberry Pi 5), and a convolutional neural network (CNN) model trained to identify morphological features and defects in Lupinus Multabilis Sweet (chocho) grains.

For the dataset determination, an average of 1,000 samples were established, covering different grain classes such as good grain, bad grain, broken grain, and other types of grains (e.g., corn or beans). The labeling of these classes was performed using open-source platforms, specifically RoboFlow, for both the labeling and dataset training processes. The next step involved training the neural network architecture, for which YOLO (You Only Look Once) in its latest version, YOLOv8, was selected.

The experimental results demonstrated a classification accuracy of 73.3%, validating the system's effectiveness in optimizing agricultural processes. This work contributes to the automation of small-scale agriculture, reducing costs and improving the quality of the final product.

Keywords: Deep Learning, Computer Vision, Raspberry, Lupinus Mutabilis Sweet, precision agriculture, ultralytics, yolov8.

Índice de Contenido

DEDICATORIA 4

AGRADECIMIENTO 5

RESUMEN 6

ABSTACT 7

Capítulo 1 17

1.1. Tema: 17

1.2. Problema 17

1.3. Objetivos 19

1.3.1. Objetivo General 19

1.3.2. Objetivos Específicos 19

1.4. Alcance 20

1.5. Justificación 22

Capítulo II 24

Fundamentación teórica 24

2. El chocho 24

2.1. Origen 24

2.2. Chocho 25

2.2.1. Taxonomía de chocho 25

2.2.2. Tipos de Chocho 26

2.2.3. Descripción de la estructura botánica 27

2.2.4. Cultivo..... 28

2.2.5. Valor Nutricional del grano de chocho (Lupinus Multabilis Sweet)..... 29

2.2.6. Cosecha del chocho (Lupinus Multabilis Sweet)..... 30

2.2.7. Comercialización 31

2.2.8. Consumo del chocho des amargado (Lupinus Multabilis Sweet)..... 31

2.3. Agricultura de Precisión 32

2.3.1. Fases de la Agricultura de precisión 32

2.3.2. Tecnología en la asociación de AP 33

2.4. Visión Artificial 34

2.4.1. Clasificación de granos con Visión Artificial..... 35

2.4.2. Procesos de la visión artificial 37

2.4.3. Aplicaciones de la Visión Artificial..... 39

2.5. Herramientas para un sistema de visión artificial..... 41

2.5.2. Hardware aplicado a la visión artificial 46

2.6. Inteligencia Artificial..... 48

2.7. Machine Learning..... 50

2.7.1. Modelos de aprendizaje automático (ML)..... 51

2.8. Deep Learning..... 54

2.8.1. Métodos de Deep Learning..... 54

Capítulo III 57

Diseño del Sistema 57

3. Metodología 57

3.1. Modelo Iterativa: Primera iteración 58

3.1.1. Requerimientos del sistema: Primera iteración 59

3.1.2. Selección de Hardware y software 64

3.2. Situación actual: Primera iteración 70

3.3. Diseño del sistema: Primera Iteración 78

3.3.1. Diseño Estructural 78

3.3.2. Descripción del funcionamiento del sistema embebido 79

3.3.3. Anotación y clase de grano (Chocho) 79

3.3.4. Chocho Maduro (grano partido) 82

3.3.5. Sistema de entrenamiento de datos 82

3.4. Etiquetado de las muestras 84

3.4.1. Etiquetado Grano Bueno 84

3.4.2. Etiquetado grano(malo) 85

3.4.3. Etiquetado grano Partido 86

3.4.4. Etiquetado otro tipo de grano 87

3.4.5. Crear Dataset 88

3.5. Diagramas del sistema 90

3.6. Conexión de Componentes 92

3.7. Conexión física de dispositivos 93

3.7.1. Diagramas de flujo 95

3.7.1.3. Flujograma de inferencia del sistema embebido. 99

3.8. Entrenamiento Red Neuronal 100

3.8.1. Script Modelo YOLO(v8) 100

Capitulo IV 103

EJECUCIÓN Y PRUEBAS DE FUNCIONAMIETO 103

4.1. Diagrama de Funcionamiento 103

4.2 Pruebas de funcionamiento 103

4.1.1. Grano Bueno 103

4.1.2. Grano bueno con otro tipo de clase 106

4.1.3. Grano Malo 106

4.1.4. Grano Partido 108

4.1.5. Prueba con diferentes clases de grano 108

4.1.6. Otro tipo de grano 109

4.3. Resultados obtenidos del sistema 110

4.4. Análisis de costo Beneficio 111

4.4.1. Costos de Hardware 111

4.4.2. Costos de software 112

4.4.3. Costo de infraestructura 113

4.4.4. Costo de ingeniería 113

4.4.5. Costo general del sistema 114

4.4.6. Beneficios 115

Conclusiones 116

Recomendaciones 117

Bibliografía 118

Anexo 124

Anexo1. Etiquetado y detección de objetos con yoloV8 124

Anexo2. Código de login al Sistema embebido 129

Anexo3. Codificación del sistema de clasificación 133

Índice de tablas

Tabla 1 *Clasificación taxonómica del chocho (Lupinus Multabilis Sweet)* *25*

Tabla 2 *Nomenclaturas* *59*

Tabla 3 *Categorías (Stakeholders)* *60*

Tabla 4 *Requerimientos Stakeholders* *60*

Tabla 5 *Requerimientos de funcionalidad* *61*

Tabla 6 *Requerimientos de Sistema* *62*

Tabla 7 *Requerimientos de arquitectura* *63*

Tabla 8 *Descripción de valores* *65*

Tabla 9 *Selección de la placa SBC* *65*

Tabla 10 *Especificaciones técnicas del Raspberry Pi5* *66*

Tabla 11 *Selección de sensor(cámara)* *67*

Tabla 12 *Especificación de las principales características del sensor* *68*

Tabla 13 *Elección de Sistema operativo* *69*

Tabla 14 *Lenguaje de Programación* *70*

Tabla 15 *Costo de Hardware* *112*

Tabla 16 *Costos de Software* *112*

Tabla 17 *Costo de infraestructura* *113*

Tabla 18 *Costos de Ingeniería* *113*

Tabla 19 *Costos generales del sistema* *114*

Índice de Figuras

Figura 1 *Maduración de la planta de chocho (Lupinus Multabilis Swett)* *30*

Figura 2 *Proceso de clasificación mediante el uso de visión artificial.* *36*

Figura 3 *Etapas de un sistema de Visión Artificial* *37*

Figura 4 *Interfaz Gráfica de Teachable Machine desde su web oficial* *45*

Figura 5 Estructura del proceso del modelo Iterativa *58*

Figura 6 *Comunidad de el abra* *71*

Figura 7 *Florecimiento de la planta de chocho* *73*

Figura 8 *Proceso de cosecha del grano de chocho* *74*

Figura 9 *Trilla del grano seco de chocho* *75*

Figura 10 *Almacenaje del grano luego de la trilla* *76*

Figura 11 *Proceso de selección de grano de chocho* *77*

Figura 12 *Simulación del diseño estructural del sistema* *78*

Figura 13 *Categoría de clasificación* *80*

Figura 14 *Grano de chocho en buen estado* *80*

Figura 15 *Conjunto bueno de grano de chocho* *81*

Figura 16 *Grano de chocho en mal estado* *81*

Figura 17 *Grano de chocho Partido* *82*

Figura 18 *Etiquetado del grano de chocho* *83*

Figura 19 *Utilización de la plataforma Robo Flow* *84*

Figura 20 *Etiquetado del grano de chocho en buena calidad* *85*

Figura 21 *Asignación de etiqueta del grano en mal estado con la clase grano malo* *86*

Figura 22 *Grano partido etiquetado en Robo Flow* *86*

Figura 23 *Proceso de etiquetado de los diferentes tipos de grano* *87*

Figura 24 *Asignación de datos para el dataset*..... *88*

Figura 25 *Datos para el entrenamiento* *89*

Figura 26 *Cambio de parámetros para generar un mayor número de dataset.* *90*

Figura 27 *Diagrama de funcionamiento del sistema* *90*

Figura 28 *Conexión de componentes electrónicos a la placa SBC.* *93*

Figura 29 *Conexión física del sistema embebido*..... *93*

Figura 30 *Diagrama lineal de funcionamiento*..... *94*

Figura 31 *Diagramas de etapas operativas del sistema embebido* *95*

Figura 32 *Flujograma de funcionamiento del sistema de etiquetado*..... *95*

Figura 33 *Flujograma de la arquitectura YOLOv8* *98*

Figura 34 *Flujograma de inferencia del sistema* *99*

Figura 35 *Creación de directorios para prueba de funcionamiento.* *100*

Figura 36 *Instalación de librerías*..... *101*

Figura 37 *Cambio de directorio para prueba*..... *101*

Figura 38 *Descarga de Yolov8 desde el repositorio* *102*

Figura 39 *Versión de Yolov8 descargado*..... *102*

Figura 40 *Identificación del grano bueno*..... *104*

Figura 41 *Grano de chocho identificado como grano bueno* *105*

Figura 42 *Precisión de reconocimiento del gano* *105*

Figura 43 *Grano de chocho en diferentes clases* *106*

Figura 44 *Identificación de grano malo de chocho* *107*

Figura 45 *Grano malo detallado según su precisión*..... *107*

Figura 46 *Detalles del grano partido 108*

Figura 47 *Determinación de grano bueno y malo ya con las clases de grano asignado 109*

Figura 48 *Identificación de otro tipo de grano 110*

Figura 49 *Detalles de validación de datos 111*

Capítulo 1

En el presente capítulo se procede a determinar las bases de realización del proyecto de titulación llamada sistema embebido de Diseño de un Sistema Embebido de Clasificación Del Grano De Lupinus Mutabilis Sweet (Chocho) Mediante el uso de Visión Artificial, describiendo la problemática, los objetivos que conlleva y su alcance.

1.1. Tema:

Diseño de un sistema embebido de clasificación del grano de Lupinus Mutabilis Sweet (chocho) mediante el uso de visión artificial

1.2. Problema

La producción del Lupinus Mutabilis Sweet (chocho andino o como tauri) conocido en las comunidades andinas, es uno de los productos agrícolas que se han venido cultivando desde tiempos ancestrales, su producción en el ecuador se lo realiza en zonas centrales como Cotopaxi, Chimborazo, Tungurahua, Bolívar y partes de las provincias de Imbabura y Pichincha. Para determinar un grano de calidad se han implementado técnicas para el mejoramiento de los sembríos del Lupinus Mutabilis Sweet (chocho), a través de charlas técnicas impartidas por el Magap (Magap, 2020), con lo que se ha pretendido mejorar la calidad en la producción del chocho. (Morán et al., 2021)

Actualmente en el Ecuador el consumo del Lupinus Mutabilis Sweet (chochos) es de 8 kg por persona anualmente (Revista Líderes, 2019), esto se debe por los beneficios, cualidades nutricionales que presenta este tipo de grano, por lo cual se ha tenido un aumento de la demanda

del consumidor en obtener este producto, por lo cual exige al agricultor producir el grano en mayor cantidad y en calidad para satisfacer el mercado de la demanda en el consumo de este producto(Revista Qualitas, n.d.), no solo en Ecuador el grano de chocho es consumida sino que también en otras partes del mundo especialmente en Alemania el porcentaje de crecimiento en exportación del grano desde los años 2013-2017 ha sido en promedio de 10%.

En la comunidad de el Abra perteneciente a la parroquia rural de Santa María, esta comunidad se ubica al sur oeste del cantón Ibarra, en esta comunidad rural los moradores se dedican a la siembra del Lupinus Mutabilis Sweet (chocho), la siembra del grano de chocho se lo realiza en suelos aproximadamente 2600 a 3400 msnm,(Alfonso F. Guzmán, 2019), la producción de esta leguminosa es aproximadamente de 15 q/a, la misma que requiere un tiempo de 10 a 12 meses aproximadamente en proceso de maduración y cosecha del grano Lupinus Mutabilis Sweet (chocho), por lo que este grano se lo puede conseguir por temporadas, es decir que no está todo el tiempo en el mercado para su comercialización.

El proceso de selección del grano del Lupinus Mutabilis Sweet (chocho) post cosecha se realiza de forma manual(De et al., 2019), en este se procede a identificar la calidad del grano para separar el grano bueno del resto, el grano es almacenado en costales como contenedores de los cuales de los pone en una mesa como montículos o se amontona en un sector de la mesa de selección de allí cada persona se ubica alrededor del montículo del grano para empezar a realizar la selección de este, este trabajo requiere de varias personas para identifica grano por grano, se ayudan de sus dedos para separar del montículo del grano uno por uno, el grano en malas condiciones se lo separa en un recipiente en el que se almacena todos los granos en mal estado, por lo cual su clasificación requiere de mucho tiempo, el tiempo de selección depende de la

cantidad de trabajadores o personas que intervengan en este proceso por lo general en un 1qq de grano de chocho con de tres personas se demora aproximadamente 4 horas o más.

Bajo esta premisa se pretende sistematizar el proceso de selección del grano de chocho por lo que se procura desarrollar un sistema de clasificación del grano Lupinus Mutabilis Sweet (chocho) con el que se procederá a identificar la calidad de este mediante la forma y el color del grano Lupinus Mutabilis Sweet (chocho), para lo cual se aplicara herramientas de software de Deep Learning enfocados a la visión artificial que aportan con técnicas de identificación de objetos de acuerdo parámetros establecidos, con lo que el agricultor se reducirá el tiempo empleado en la selección del grano.

1.3.Objetivos

1.3.1. Objetivo General

Diseñar un prototipo para la clasificación del grano de Lupinus Mutabilis Sweet (chocho) postcosecha, basado en técnicas de visión artificial, con el fin de optimizar el tiempo empleado en la selección de la calidad del grano del chocho.

1.3.2. Objetivos Específicos

- Revisar la bibliografía en fuentes fiables referente al tema planteado que permita consolidar la base teórica y el estado del arte en desarrollo del proyecto.
- Diseñar el prototipo de clasificación de grano de Lupinus Mutabilis Sweet (chocho) postcosecha aplicando métodos de visión artificial, mediante la integración de hardware, software en el proceso de identificación y selección del grano.
- Emplear algoritmos de Deep Learning que permita el entrenamiento del prototipo de clasificación del grano Lupinus Mutabilis Sweet (chocho), el cual luego de su

entrenamiento se procederá a exportar hacia una placa embebida para su ejecución y aplicación

- Realizar pruebas de funcionamiento del prototipo en un ambiente de prueba durante un tiempo determinado, en el que se evidencie los resultados brindados por el prototipo de clasificación del grano Lupinus Mutabilis Sweet (chocho), para establecer su margen de error y tiempo de ejecución con el método de conteo manual del grano.

1.4.Alcance

El presente proyecto pretende brindar una facilidad al agricultor en realizar el proceso de selección de la calidad del grano de Lupinus Mutabilis Sweet (chocho) para comercializar el producto en el valor del quintal ya establecido (del Salto Pazmiño, 2019), uno de los inconvenientes es que al no tener una selección de calidad del grano de Lupinus Mutabilis Sweet (chocho) su valor se deprecia en el mercado. Por lo cual el proyecto se ajusta a un prototipo de clasificación mediante la visión artificial que posibilite identificar de manera autónoma la calidad del grano de chocho. Por lo que la metodología que se aplicara en el desarrollo de este proyecto es la metodología interactiva.

En la primera etapa se procede a identificar los requerimientos generales que tiene el prototipo con el sistema de clasificación, este se basará en la revisión de referencias bibliográficas fiables y actuales, que permita establecer un criterio técnico en el desarrollo de cada etapa que tenga el proyecto y de esta manera estructurar el diseño del prototipo adecuadamente desde su etapa inicial.

Como segunda etapa se procede a realizar un armado del prototipo electrónico del sistema de clasificación del grano de Lupinus Mutabilis Sweet (chocho) post cosecha, para lo cual se

identifica placas embebidas que permitan trabajar tecnología IA por consiguiente se establece una selección detallada de características, en dispositivos tecnológicos por lo que se establece una comparativa de hardware en procesamiento y capacidad de almacenado de información para la implementación en el prototipo del sistema embebido.

Tercera parte corresponde a la etapa de codificación por lo que se aplicara herramientas predefinidas de Deep Learning, orientados a la visión artificial(López et al., 2016), se desarrolla con lenguajes de alto nivel como es Python por su versatilidad en librerías, bibliotecas y frame-works orientados a la visión artificial,(Botto-Tobar et al., 2021) ,en este se procede a entrenar el sistema de clasificación mediante captura de fotogramas a través de una cámara digital conectado a una placa embebida, en la cual se almacenara una base de datos de entrenamiento, el proceso de gestión del prototipo en el que se pretende visualizar el procedimiento a ejecutarse es una interfaz usuario aplicando módulos de tkinter, PySimpleGUI (Podržaj, n.d.), que permiten la integración en el lenguaje de programación de alto nivel Python y de esta manera visualizar al grano de Lupinus Mutabilis Sweet (chocho), clasificado con las consideraciones de grano en este caso por su tamaño y el color.(Freire Diaz et al., 2021).

Como en etapa final de realizará el proceso prueba de funcionamiento del sistema aplicado a un ambiente de pruebas en el que el prototipo de manera autónoma deberá mostrar la capacidad de identificar el grano de Lupinus Mutabilis Sweet (chocho) de acuerdo con los parámetros establecidos en la etapa de entrenamiento, mediante este prototipo procederá a validar que sistema de clasificación identifique la calidad del grano optimizando el tiempo empleado para su selección y de esta manera el agricultor pueda comercializar su producto de calidad en un precio justo.

1.5.Justificación

De acuerdo con Plan de creación de Oportunidades ejecutada por la Secretaria Nacional de Planificación correspondiente al período 2021 - 2025; el sector agrícola rural en la cual no se ha tenido una normativa legal que regule sobre temas de inclusión, producción y comercialización de productos agrícolas, por lo consiguiente este proyecto pertenece al eje 1, Económico y Generación de Empleo, el objetivo número 3 el cual determina que: " Fomentar la productividad y competitividad en los sectores agrícola, industrial, acuícola y pesquero, bajo el enfoque de la economía circular", también con el eje 2, Social, el objetivo número 8 donde menciona que: “Generar nuevas oportunidades y bienestar para las zonas rurales, con énfasis en pueblos y nacionalidades”. Con la mención expuesta es por lo cual el presente proyecto pretende aportar a la integración de equipos tecnológicos en temas de producción agrícola con lo que permitirá al pequeño productor optimizar el tiempo empleado en la comercialización de la producción. (secretaria nacional de Planificación, n.d.)

Los aportes alimenticios que presenta el grano de Lupinus Mutabilis Sweet (chocho) des amargado para el consumo humano es beneficioso, según los técnicos del MAGAP en el ecuador el consumo de chocho es de 8 kg aproximadamente por persona anualmente; la alta demanda se debe a que la población conoce sobre las cualidades nutricionales y la versatilidad gastronómica de la leguminosa(Ecuador et al., 2019), por lo que el agricultor debe realizar un proceso adecuado para su cultivo como también para la etapa de cosecha, la selección del grano es fundamental para determinar la calidad del Lupinus Mutabilis Sweet (chocho) para su comercialización.

Según (García, 2017), con el sistema de clasificación del grano de Lupinus Mutabilis Sweet (chocho) a través de herramientas de Deep Learning orientados a la visión artificial(Constante et al., 2016) se pretende disminuir el tiempo empleado en la selección del grano de Lupinus Mutabilis

Sweet (chocho) post cosecha, ya que el agricultor realiza el proceso de selección de forma manual, es decir grano por grano lo cual requiere emplear mucho tiempo para obtener un producto seleccionado de calidad.(De & Agrarias, n.d.) Al aplicar este sistema de clasificación del grano de Lupinus Mutabilis Sweet (chocho) post cosecha permitirá al agricultor optimizar el tiempo en la selección del grano y obtener un grano clasificado de calidad que le permitirá comercializarlo a un precio justo.

En la actualidad en áreas de agricultura no se ha tenido un desarrollo importante en cuanto a tecnología ya que este campo presenta diferentes tipos de ambientes en los que el agricultor trabaja, con la inclusión de tecnología en el agro no solo se tendría buenos resultados en los productos sino que también se optimizaría el tiempo empleado en su producción y comercialización.(Kamilaris & Prenafeta-Boldú, 2018) En el caso de procesos de Deep Learning (aprendizaje profundo) constituye una técnica reciente y moderna para el procesamiento de imágenes y el análisis de datos, con resultados prometedores y un gran potencial.(Kamilaris et al., 2017)

Capítulo II

Fundamentación teórica

En este capítulo se presenta el desarrollo del marco teórico con la temática relacionada para el desarrollo del proyecto, la estructura bibliográfica se inicia con la introducción del origen del chocho (Lupinus Multabilis Sweet) como sus características generale, su producción y comercialización. Como segunda parte se tiene la descripción de generales de la visión artificial su aplicación en diferentes áreas en la industria. Siguiendo con la estructura del marco teórico en la tercera parte se tiene las herramientas tanto de software y hardware aplicados a la visión artificial y finalizando con descripción de aprendizaje profundo y sus técnicas.

2. El chocho

Es una leguminosa de la familia del Lupino (Lupino Multabilis Sweet), es un grano de consumo humano que brinda un porcentaje considerables de nutrientes, este grano es originario del Perú, en el Ecuador se lo puede encontrar en la serranía principalmente en provincias de Cotopaxi, Tungurahua, Bolívar y partes de Imbabura. La comercialización de este grano se lo realiza en grano seco, en grano des amargado o en harina de chocho para el consumo humano.

2.1.Origen

Según un estudio realizados por (Chalampuente Doris et al., 2021), el Lupinus Mutabilis Sweet, también conocido como altramuz o tarwi andino, tuvo su origen en la época incaica, específicamente en la cultura Nazca, y se encontraron restos del grano representados o tallados en cerámica en la cultura Tiahuanaco en el Perú. Varias investigaciones muestran el origen de esta planta que se encontró en la región andina de Bolivia, Ecuador y Perú, donde existe la mayor variabilidad genética. El tarwi andino o altramuz es una especie que presenta una amplia diversidad

genética y la gran capacidad de adaptación a diferentes condiciones de suelos, precipitación, temperatura y altitud. En la región andina se han identificado 83 especies de tarwi ya clasificados por su nombre científico en el género Lupinus, según las investigaciones realizadas en ese tiempo. (Cabrera Ovando et al., 2020)

2.2.Chocho

El género Lupinus, perteneciente a la familia Fabaceae, cuenta con una diversidad de especies de las cuales ha prevalecido solamente cinco tipos de plantas para su cultivo de las 83 especies que se había identificado previamente, entre las cuales se tiene: el Lupinus albus, Lupinus angustifolius, Lupinus luteus, Lupinus Mutabilis, y Lupinus consentenii. De estos cinco tipos de plantas se identifica al grano de Lupinus Mutabilis Sweet, conocido como chocho, es una leguminosa que ofrece un alto índice de valor nutricional debido a su contenido de proteínas (entre 20 y 46% en grano seco), carbohidratos, minerales y fibra.

2.2.1. Taxonomía de chocho

A continuación, en la tabla 1 se muestra la taxonomía general del chocho, en esta se especifica a que tipo de subespecie pertenece el chocho (Lupinus Mutabilis Sweet).

Tabla 1

Clasificación taxonómica del chocho (Lupinus Multabilis Sweet)

Taxonomía	
Orden	Fabales
Suborden	Leguminosae
Familia	Fabaceae
Subfamilia	Faboideae
Tribu	Genisteae
Género	Lupinus
Especie	Multabilis Sweet

Nota: Datos tomado desde (Vaca M, 2021)

2.2.2. Tipos de Chocho

La planta de Lupinus Mutabilis Sweet presenta diversas variabilidades morfológicas que hacen difícil identificación es decir a que tipo de especie pertenece para luego establecer la clasificación y su descripción, por lo cual de los cinco tipos de plantas que de identifico se detalla tres subtipos principales de especies que en los últimos años se han cultivado en regiones de Perú, Ecuador, Bolivia y parte de Colombia y los tipos restantes ya se han ido desapareciendo tras no ser cultivados, a continuación, se detalla a cada una de estas especies:

- **Lupinus Mutabilis**: Lupino este subtipo de los a identificado en cembrios al norte de Perú y partes del Ecuador, de detallan por sus características como: contar con una ramificación más prolífica, muy tardía, tiempo de maduración extensa, mayor vellosidad de hojas y tallos, algunos eco tipos se comportan como bienales, tolerantes a la antracnosis.(Chalampuente Doris et al., 2021)
- ***Lupinus Mutabilis*, tarwi**: Este subtipo se encuentra establecido en el centro y sur del Perú, la diferencia se distingue en el tipo de ramificación que presenta la cual apenas es ramificado, moderadamente tardío, su maduración no requiere de mucho tiempo, algo tolerante a la antracnosis.(Chalampuente Doris et al., 2021)
- ***Lupinus Mutabilis*, tauri**: Este subtipo se encuentra asentado en las tierras altas de Perú y Bolivia, el tamaño de la planta es más pequeño va desde 1metro hasta los 1,40 m de altura con un tallo principal desarrollado, muy temprano su maduración se lo hace en muy poco tiempo, susceptible a la antracnosis.(Chalampuente Doris et al., 2021)

2.2.3. Descripción de la estructura botánica

La planta del Lupinus Mutabilis Sweet (chocho) su estructura física depende del tipo de suelo, ambiente y del estado de crecimiento de la planta, por lo general en proceso de crecimiento, desarrollo y maduración se tiene la siguiente estructura física que se describe a continuación:

- **Hojas**. La hoja de Lupinus es de forma digitada (se asemeja a los dedos de una persona con la mano abierta), generalmente compuesta por ocho folíolos que varían entre ovalados a lanceolados. En parte inferior del pecíolo existen pequeñas hojas estipulares. El color puede variar dependiendo la etapa en que se encuentre la planta del chocho de amarillo verdoso a verde oscuro, dependiendo del contenido de antocianina, es decir el color de las hojas varían constantemente desde proceso de crecimiento hasta llegar a su etapa de maduración.(Sánchez,2017)
- **Flores e inflorescencia**. El Lupinus Mutabilis Sweet (chocho) presenta una corola grande de 1 a 2 cm, con cinco pétalos y compuesta por un estandarte, dos quillas y dos alas. Se ha detallado que en una sola planta de chocho pueden existir aproximadamente 1000 flores. Cuando la coloración de la flor es morada este se inicia de formación de vainas en el que se contiene el grano de chocho hasta la maduración en el que se evidencia un azul claro hasta uno muy intenso y de allí se origina su nombre científico, Mutabilis, es decir que cambia. Los colores más comunes son los diferentes tonos de azul e incluso púrpura; menos frecuentes son los colores blancos, crema, rosado y amarillo. ("Características del Tarwi - WordPress.com")(Sánchez, 2017)
- **Semilla**. Las semillas del Lupinus Mutabilis Sweet (chocho), es una vaina de 5 a 12 cm con una variación dependiendo al crecimiento de la planta, miden entre 0,5

a 1,5 cm. Un kilogramo tiene 3500 a 5000 semillas. La variación en tamaño depende tanto de las condiciones de crecimiento como del eco tipo. (“Proyecto Semilla de Palta | PDF | Aguacate | Alimentos - Scribd”) La semilla está recubierta por un tegumento endurecido que puede constituir hasta el 10% del peso total. Los que se aprecian en el grano seco van desde blanco, amarillo, gris, ocre, pardo, castaño, marrón y colores combinados como marmoleado, media luna, ceja y salpicado, hasta granos que están deformados.(Sánchez, 2017)

- **Tallo.** La altura de la planta se clasifica por el eje principal que varía entre 0,5 a 1,30 m esto es una altura referencial. El tallo de Lupinos Mutabilis (chocho) es generalmente muy leñoso y ramificado el mismo que se puede utilizar como combustible. El tallo al estar seco es posible emplear como material de combustión.(Sánchez, 2017)
- **Ramificaciones**. Las ramificaciones de la planta de chocho varían según como fue el crecimiento es decir si se tuvo un cuidado continuo las ramificaciones serán abundantes. Según el tipo de ramificaciones, desde la planta se puede determinar si es de eje central predominante, con ramas que inician desde el tronco o el tallo principal de la planta, tipo candelabro, o ramas terminales; o de una ramificación desde la base con la misma altura. El número de ramificaciones en una planta son aproximadamente de 52 ramas.(Sánchez, 2017)

2.2.4. Cultivo

Con la domesticación del cultivo del grano de Lupinus Mutabilis Sweet se ha perdido un número importante de tipos de plantas, en la región andina en donde se practica su cultivo se han identificado alrededor de unas 85 tipo de Lupinus de los cuales 5 fueron adaptados para su cultivo

pero en la actualidad se tiene solamente 3 subtipos de plantas que predominan en el cultivo en partes de Bolívar, Perú y Ecuador , los parientes silvestres que muestran diversidad y variabilidad que se encuentran clasificados en otro tipo de subespecie que es el altramuz andino.

En el Ecuador el cultivo de esta leguminosa se lo practica en terrenos áridos o con muy pocos nutrientes con un tipo de clima frio, la planta del lupinos no requiere de muchos nutrientes en el suelo en procesos de crecimiento, la distribución geográfica para su cultivo hace referencia la parte norte, centro y sur del territorio ecuatoriano, empezando desde las provincias de Carchi, Imbabura hasta Cotopaxi y partes de Bolívar.

En la zona norte del país el cultivo de chocho está presente en algunos cantones de las provincias de Imbabura y Carchi, según datos del MAGAP (2016), según la distribución de zonificación agroecológica realizados, en estas provincias se tienen una potencialidad elevada para el cultivo de esta leguminosa particularmente los cantones de Otavalo, Ibarra y Antonio Ante (Imbabura) y Tulcán (Carchi). Según el proyecto SAN IMBABURA (2017) en Imbabura había 118 productores de chocho distribuidos en los cantones Cotacachi, Urcuquí, Pimampiro, Otavalo e Ibarra. (Basantes et al., 2022)

2.2.5. Valor Nutricional del grano de chocho (Lupinus Multabilis Sweet)

El grano de chocho presenta valores nutricionales en gran cantidad aportando al consumidor, para el consumo humano el grano debe pasa por un proceso de des amargo es decir se debe cocinar para quitar el sabor amargo que tiene el grano seco, el grano de chocho Lupinus Mutabilis Sweet es amargo debido a la presencia de alcaloides quinolizidínicos por lo que se debe preservar en agua limpia el grano ya cocinado, el grano contiene en promedio 42% de proteína, en base seca; sin embargo, el proceso de des amargo (eliminación de alcaloides) permite concentrar aún más el contenido de este nutriente, registrando valores de hasta 51%, en base seca. El grano

de chocho cuenta con gran cantidad de aceite desde 18 a 22%, en el que se detalla los siguientes ácidos grasos (Quishpe Defaz, 2020)

- Oleico: 40,40%
- Linoleico (6): 37,10%
- Linoleico (3): 2,90%

2.2.6. Cosecha del chocho (Lupinus Multabilis Sweet)

Este grano tiene una etapa de crecimiento, desarrollo para luego entra en el proceso de maduración en que la planta del chocho se empieza a secarse desde las hojas hacia las vainas. Una vez completada la maduración y cuando las vainas adquieren una coloración amarillenta, las plantas son arrancadas manualmente y colocadas en un contenedor o saco para luego ser transportado a un sitio seco libre de humedad para realizar el proceso de la trilla. La trilla es el proceso que más tiempo demanda; ya que se utiliza una maquina trilladora de granos en el que se realiza la separación de la cascara del grano del chocho.(Tapia, 2015). En la Figura 1 se detalla el estado de una vaina en la etapa de maduración del grano de chocho.

Figura 1

Maduración de la planta de chocho (Lupinus Multabilis Swett)

Nota: Vaina de la planta de chocho en proceso de maduración. Tomado de (Juncosa et al., 2022)

2.2.7. Comercialización

El grano de chocho seco se lo comercializa dependiendo a la calidad del grano, para obtener una mayor remuneración es necesario realizar un proceso de selección previo para descartar impurezas o granos de chocho en mal estado.

Según (Basantes et al., 2022), el chocho se comercializa tanto como grano amargo es decir el grano en seco o des amargado cuando el grano esta ya cocida, en el segundo caso del producto se vende en mercados y bodegas en las ciudades de Latacunga, Riobamba, Otavalo, Ibarra y Ambato; mientras que el grano amargado es comercializado por intermediarios, el sistema de comercialización de los centros de producción se realiza a través de intermediarios o acopiadores zonales, que estos a su vez entregan el grano a los procesadores donde ejecutan el proceso de des amargado. Los procesadores distribuyen el chocho a vendedores o tiendas hasta llegar al consumidor final.

El valor comercial actual en el mercado nacional para el grano de chocho se lo establece por quintal clasificado, siendo un valor mínimo por quintal de 75 USD hasta llegar a un máximo de 110 USD por quintal. Para el año 2017 y con base al fomento, promoción del cultivo, el precio del quintal de chocho alcanzó precios históricos de 180 dólares. El precio de pago por los intermediarios en esa fecha estaba entre 75 a 110 USD q. Para los siguientes años se evidencia que los precios al productor bajaron a 80 y 90 USD q y en el año 2021 no se evidencia una mejora o alza en el valor comercial del q de chocho; el precio se ha mantenido relativamente estable entre 75 a 110 USD q dependiendo de la temporada.(Basantes et al., 2022)

2.2.8. Consumo del chocho des amargado (Lupinus Multabilis Sweet)

En el ecuador el consumo del chocho des amargado ha tomado un incremento considerable con respecto a otros países. Actualmente en el Ecuador el consumo del Lupinus Mutabilis Sweet (chochos) es de 8 kg por persona anualmente (Revista Líderes, 2019), por los beneficios,

cualidades nutricionales que presenta este tipo de grano, por lo cual se ha tenido un aumento de la demanda del consumidor en obtener este producto y por lo cual exige al agricultor producir en mayor cantidad para satisfacer el mercado de la demanda en el consumo de este grano.

Según (MAGAP, 2022), el grano presenta propiedades nutritivas del chocho, leguminosa de alto valor nutritivo, que se distingue por su contenido de proteína, El mineral predominante en el chocho es el calcio, con una concentración promedio de 0,48%. Este componente, es una sustancia blanquecina que los dientes y huesos deben absorber para asegurar un crecimiento adecuado y mantener la solidez.

2.3.Agricultura de Precisión

La agricultura de precisión es la implementación de tecnologías en procesos de producción en diferentes cultivos. Es un sistema desarrollado en analizar y controlar la variación espaciotemporal del terreno y el cultivo. La variación espacial comprende las diferencias en fertilidad de distintas secciones del terreno y las que se dan en el crecimiento de las plantas cultivadas. La variación temporal engloba las diferencias observadas en la producción de un mismo terreno entre una temporada y otra.

2.3.1. Fases de la Agricultura de precisión

Para complementar el funcionamiento de las fases de la agricultura de precisión es fundamental la complementación de tecnologías, equipos o dispositivos asociados en adquisición, procesamiento y ejecución de información adquirida.

La agricultura de precisión como cualquier proceso requiere de fases para su operación por lo que se define tres fases principales:

- Recolección de datos
- Análisis de datos

- Implementación

2.3.2. Tecnología en la asociación de AP

De manera principal para un sistema de AP existen cinco tecnologías asociadas a la AP: los sistemas de posicionamiento global (GPS) y de información geográfica (GIS), sensores remotos, monitores de rendimiento u aplicación y maquinaria inteligente. Cada una de estas tecnologías brindan información que es requerida en la siguiente fase o proceso.

2.3.2.1.GPS

Una tecnología principal en un AP es los sistemas de posicionamiento global (GPS), la cual ha permitido registrar la variabilidad espaciotemporal y controlar con exactitud geográfica la maquinaria agrícola controlados remotamente. Éste se emplea en monitores de rendimiento de equipos agrícolas, pilotos automáticos o en equipos de aplicación variable.

2.3.2.2.GIS

Los GIS son sistemas informáticos usados para almacenar, visualizar y analizar datos referidos geográficamente. En la AP permiten analizar la información obtenida mediante los distintos receptores (v.g. sensores remotos), para tomar decisiones sobre el manejo de la variabilidad espaciotemporal.

2.3.2.3.Sensores Remotos

Los sensores remotos son sistemas (satelitales o portátiles) que obtienen información del cultivo, sin tener un contacto físico con éste. Se emplean en la recolección de datos sobre la administración del agua de riego, contenido de materia orgánica, vigor de las plantas (por ejemplo, su contenido de clorofila), enfermedades vegetales, plagas, mapeo de malezas, sequía e inundaciones. Para ser eficientes, deben estar bien calibrados y poseer suficiente resolución.

2.3.2.4.Monitores de rendimiento y aplicación

Los monitores de rendimiento obtienen información sobre la cantidad (granos recolectados por unidad de tiempo) y la calidad del cultivo (entre otras características, la humedad del producto).16 Otro tipo de monitores, los de aplicación variable, se usan para dosificar la cantidad de insumos por cada sección del terreno, por ejemplo, la dosis de semilla o agroquímicos.

2.3.2.5.Maquinarias Inteligentes

La cosecha de algunos productos, por ejemplo, frutas, requiere una intensa labor manual por parte de trabajadores temporales, lo que incrementa los gastos de producción. Se han desarrollado cosechadoras inteligentes capaces de diferenciar frutos maduros e inmaduros. También existen sistemas de detección de flores en árboles frutales, que permiten estimar la producción, la aplicación variable de agroquímicos y la recolección automatizada de frutos. Además, existen sistemas de piloto automático que controlan la maquinaria agrícola vía DGPS. La investigación básica y aplicada en inteligencia artificial impactará positivamente a la AP.

2.4.Visión Artificial

La visión artificial, o también denominada visión por computador, es considerada una rama de la inteligencia artificial que tiene como objetivo principal es que las máquinas sean capaces de interpretar el entorno que las rodea tal y como lo realiza un ser humano mediante el sentido de la vista.(Sevillano-Bustos, 2015)

La visión artificial es una forma de aprendizaje profundo en la que los objetos se identifican directamente a partir de aplicar el proceso de pixelado de imagen sin procesar. Como primera etapa se realiza el etiquetado de cada imagen con el diagnóstico correcto, representando lo que se describe como "verdad básica". La visión artificial o visión por computadora se centra en el reconocimiento de imágenes y videos que maneja las tareas asignadas, como la clasificación,

detección e interpretación de objetos, para categorizar los resultados predefinidos. (Khemasuwan et al., 2021)

La visión artificial comprende métodos, técnicas a través de los cuales los sistemas de visión artificial pueden desarrollarse y ejecutarse de manera razonable en aplicaciones prácticas. Esta área de la informática incluye el software, el hardware y las técnicas de imagen necesarias para esos métodos. Básicamente, un sistema de visión artificial se compone de las siguientes etapas: adquisición de imágenes, procesamiento de imágenes, segmentación y clasificación. (Lu et al., 2017)

2.4.1. Clasificación de granos con Visión Artificial

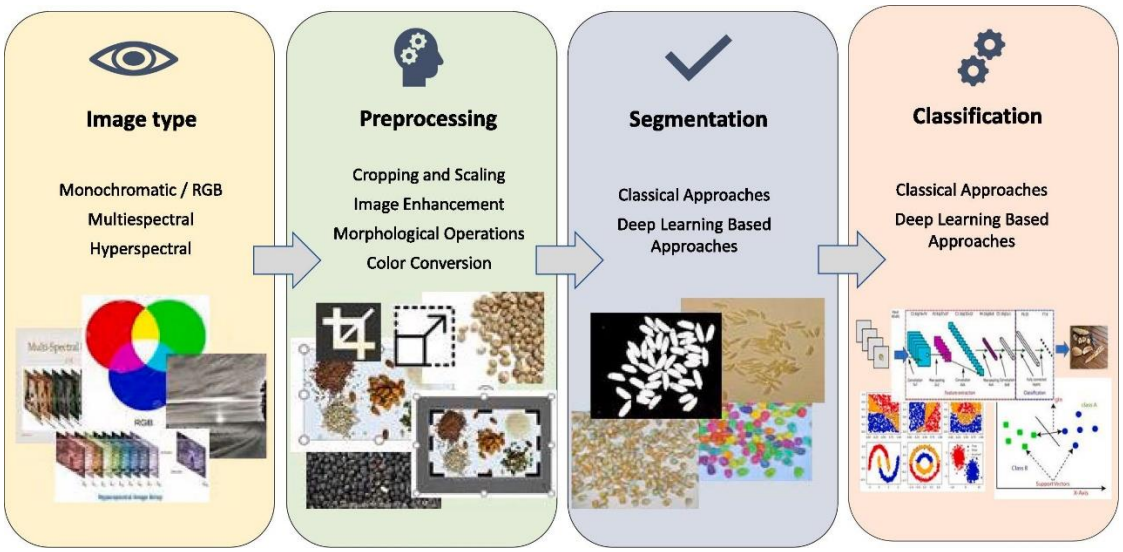
En sistema de clasificación de granos a través de visión artificial es la implementación de varias etapas que se ejecutan siguiendo un orden ya establecido, es muy importante identificar qué tipo de grano a identificar. De tal manera se estable 4 etapas principales con lo que un sistema de visión artificial es integrado en procesos de selección de granos. A continuación, se identifica las 4 etapas principales:

- Adquisición de imágenes
- Procesamiento
- Segmentación
- Clasificación

En la figura 2 se muestra el proceso que se realiza en un sistema de clasificación de grano imprentando técnicas de visión artificial.

Figura 2

Proceso de clasificación mediante el uso de visión artificial.



Nota: Sistema de visión artificial para la clasificación de grano. Tomado de (Javier et al., n.d.)

2.4.1.1.Adquisición de imágenes

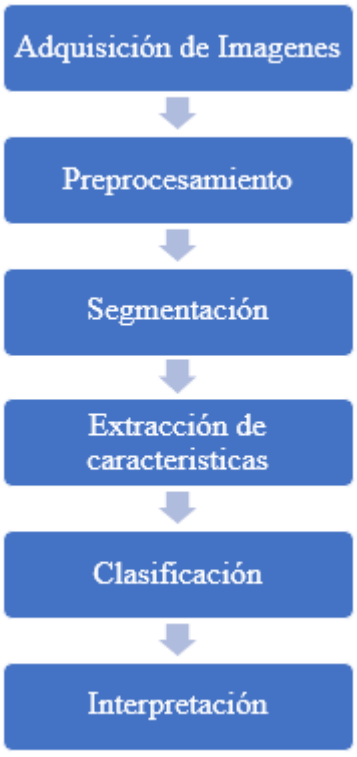
La primera etapa del sistema general está relacionada con la adquisición de imágenes. Aunque las imágenes del espectro visible (por ejemplo, imágenes monocromáticas, imágenes en color representadas en el espacio de color RGB) generalmente se consideran como entrada por los sistemas de clasificación de granos alimenticios, hay varias obras dedicadas a procesamiento de imágenes de otras bandas espectrales, por ejemplo, espectro infrarrojo. Además, algunos enfoques explotan otros tipos de imágenes (por ejemplo, multiespectrales o hiperespectrales), cubriendo no solo el espectro visible sino también las bandas espectrales ultravioleta o infrarroja. Este tipo de imágenes nos permiten obtener información útil para el proceso de clasificación, que no está disponible en el dominio clásico de banda única. Esta sección revisa los enfoques de vanguardia en bandas espectrales visibles e infrarrojas junto con enfoques multiespectrales e hiperespectrales, destacando sus limitaciones, ventajas e inconvenientes.

2.4.2. Procesos de la visión artificial

En un sistema de visión artificial o visión por computadora se puede identificar varias técnicas que tienen un orden o secuencia, tales como el procesamiento de imágenes (captura, transformación, codificación de imágenes) o como el reconocimiento de formas (teoría estadística de decisiones, enfoques sintácticas y neuronales aplicados a la clasificación de patrones). En la Figura 2 se muestra un proceso en bloques que detalla las etapas en un sistema de visión artificial este tipo de sistemas, además se incluyen técnicas de modelado geométrico y procesos de conocimiento. (SOBRADO, 2020)

Figura 3

Etapas de un sistema de Visión Artificial



Nota: Procesos que permite la implementación de un sistema de visión artificial. Tomado de (Javier et al., n.d.)

A continuación, se detalla a cada uno de los procesos que debe llevar a cabo la visión artificial para interpretar una imagen o un video:

1. **Adquisición de Imágenes:** Este es el primer procedimiento que se realiza en un sistema de visión artificial, la cual comprende en obtener imágenes digitales a través de fotografías o videos que posteriormente van a ser procesados, en esta etapa es fundamental el uso de cámaras, sensores o cualquier otro dispositivo que pueda capturar datos visuales.
2. **Preprocesamiento:** Luego de obtener los datos de inicio, es necesario aplicar técnicas de corrección, es decir quitar el color, mejorar la imagen o reducir el ruido y distorsiones que puede tener la imagen, este proceso de lo realiza mediante aplicaciones específicas.
3. **Segmentación:** En esta etapa es fundamental la identificación del área o regiones de las imágenes ya procesados. Esto puede implicar separar los objetos de un fondo, detectar bordes o identificar patrones o características específicas de la imagen.
4. **Extracción de características:** La extracción de características es posible establecer luego de la segmentación de la imagen, en este se extraen características específicas de las regiones identificadas. Esto puede incluir información sobre la forma, el color, la textura u otras características que puedan ser relevantes para la tarea específica que se está realizando.
5. **Clasificación:** Como parte de la etapa final es necesario asignar etiquetas a las regiones de interés en función de las características extraídas. Esto puede involucrar el uso de algoritmos de aprendizaje automático o redes neuronales para clasificar las imágenes en diferentes categorías.

6. **Interpretación:** En la última etapa, se interpreta el resultado final del procesamiento de la imagen para obtener información útil o tomar decisiones. Esto puede incluir la identificación de objetos en la imagen, la detección de patrones o la toma de decisiones en función de la información obtenida.

2.4.3. Aplicaciones de la Visión Artificial

Los sistemas de visión artificial es una tecnología en constante desarrollo desde su inicio hasta la actualidad de manera que se ha mejorado su utilización y aplicabilidad en diferentes áreas, siendo un sistema de aporte fundamental en procesos o en control de calidad por lo que hoy en día es una de las áreas que mayor investigación.

La tecnología de visión artificial o visión por computadora es una herramienta que se basa en la innovación constante con el desarrollo de varios métodos tecnológicos, como la inteligencia artificial y el procesamiento de imágenes. La implementación de esta tecnología en el proceso de producción no solo puede mejorar la calidad o la eficiencia de procesos, sino que también a tener un control en el diseño y procesamiento del producto, con lo que también facilita integrar la innovación y controlar en el contenido de la información de producción o producto. En la sociedad actual, la tecnología de visión artificial se ha convertido en una herramienta indispensable, que influye en gran medida en la vida cotidiana y el trabajo de las personas, por lo que su estudio resulta de gran importancia.(Li & Zhang, 2020). A continuación, se muestra las diferentes áreas en las que más se destaca la utilización de la visión artificial.

2.4.3.1.Sistemas de visión artificial en la Medicina

En el campo médico, siempre ha sido un área clave para la investigación, el desarrollo y la aplicación de alta tecnología. Hoy en día, en las ciencias médicas, la aplicación de la tecnología

de visión artificial es cada vez más amplia. Por lo que las aplicaciones parte en adquirir imágenes de pacientes con dolores de cabeza hasta pacientes con enfermedades catastróficas.(Esteva et al., n.d.) Por lo tanto, la visión artificial tiene muchas aplicaciones en el campo de la medicina como:

- **Diagnóstico médico:** Mediante la visión artificial es posible analizar imágenes médicas, como radiografías, tomografías y resonancias magnéticas, para detectar enfermedades y lesiones en el cuerpo humano.
- **Detección de cáncer:** Aplicando técnicas de visión artificial se puede utilizar para determinar imágenes de tejido humano con lo que posibilita detectar la presencia de células cancerosas.
- **Cirugía robótica:** En cuanto a la cirugía robótica utilizando la visión artificial para proporcionar imágenes de alta resolución del interior del cuerpo del paciente para que el cirujano pueda realizar una cirugía precisa.

2.4.3.2.Sistemas de visión artificial en la agricultura

La visión artificial en la agricultura ha tenido un desarrollo constante ya que mediante métodos de visión artificial es posible determinar la calidad de los cultivos, monitorear un ambiente dentro de un invernadero o también para realizar la clasificación de la calidad de productos. Por lo tanto, en el campo agrícola el uso de herramientas tecnológicas hace que se tenga una eficiencia en labores agrícolas.

La visión artificial en la agricultura se refiere al uso de técnicas de aprendizaje automático y de redes neuronales para detectar y monitorear cultivos, ganado, fincas y jardines. Los agricultores pueden utilizar esta tecnología para monitorear el crecimiento de los cultivos, detectar enfermedades, plagas y malezas, estimar el rendimiento de los cultivos, realizar el riego de

precisión, monitorear la condición del suelo, guiar equipos autónomos, utilizar software de gestión de fincas y asegurar la calidad de los alimentos. La visión artificial en la agricultura puede ayudar a los agricultores a tomar decisiones informadas para mejorar sus rendimientos y administrar sus fincas de manera sostenible.(Hsu et al., 2020)

2.4.3.3.Visión artificial en la aplicado a la automatización Industrial

En la sociedad actual, los requisitos de las personas para la eficiencia y la calidad de la producción industrial mejoran constantemente, por lo que la aplicación de la tecnología de visión artificial en esta industria está más extendida. Hoy en día, la tecnología de visión artificial es ampliamente utilizada en las industrias de impresión y embalaje de productos. A través de la aplicación de la tecnología de visión por computadora, la eficiencia de la producción industrial se ha mejorado significativamente y, al mismo tiempo, se ha mejorado la calidad de los productos El uso de la tecnología de visión por computadora puede lograr el objetivo de adquisición de imágenes en poco tiempo, capturar con precisión la imagen del momento en que se deja caer la moneda y transmitir la información de la imagen al sistema informático a través de la red de relaciones relevante, y luego usar el reconocimiento, El poder de procesamiento capta con precisión la calidad de las monedas. En la actualidad, la tecnología de visión artificial también puede medir con precisión las dimensiones de varias piezas y calcularlo a través de la computadora para hacer que la información del contorno de la pieza detectada sea más precisa, a fin de lograr una medición precisa del tamaño de la pieza.(Zhang et al., 2020)

2.5.Herramientas para un sistema de visión artificial

Las herramientas que se utilizan para un sistema de visión artificial se dividen en dos grupos un grupo de herramientas de software enfocado a visión artificial y el segundo el hardware

que permita la implementación se sistema, por lo tanto, se procede a identificar herramientas que conforman a cada uno de estos grupos.

2.5.1.1.Software enfocado a la visión artificial

En la visión artificial en fundamental el emplear lenguajes de programación que permita realizar una secuencia de instrucciones para la adquisición, procesamiento de datos y el entrenamiento del sistema de visión artificial. Hay diferentes plataformas o lenguajes de programación que ya están predefinidos para la utilización. (Gabel, 2017). A continuación, se muestra algunas de estas herramientas:

- **OpenCV (Open Source Computer Visión Library):** Es una librería orientada al procesamiento y análisis de imágenes compatible con varios lenguajes de programación como Python, C++, Java, entre otros.
- **TensorFlow:** Esta plataforma de aprendizaje automático de código abierto desarrollada por Google. Puede ser utilizada para realizar procesamiento de imágenes y aplicarlos en sistemas de visión artificial.
- **PyTorch:** Es una biblioteca de aprendizaje automático de código abierto desarrollada por Facebook. Se utiliza principalmente para procesamiento de imágenes y visión artificial.
- **Caffe:** Es una biblioteca de aprendizaje profundo de código abierto que se utiliza para procesamiento de imágenes y visión artificial.
- **MATLAB:** Es un software de programación numérica que se utiliza para procesamiento de imágenes y visión artificial.

2.5.1.2. Python

Según Alberto Boschetti y Luca Massaron (2018) en su libro Python Data Science Essentials: “Python” es un lenguaje de programación de código abierto, orientado a objetos y multiplataforma. En comparación con algunos de sus competidores directos (por ejemplo, C ++ o Java), Python es muy conciso. Le permite construir un prototipo de software funcional en muy poco tiempo y, sin embargo, se ha convertido en el lenguaje más utilizado en la caja de herramientas del científico de datos, no solo por eso. También es un lenguaje de propósito general, y es muy flexible debido a una variedad de paquetes disponibles que resuelven un amplio espectro de problemas y necesidades.(Boschetti & Massaron, 2018)

Python es un lenguaje de programación moderno que admite estilos de programación orientados a objetos, funcionales e imperativos. Es ideal para iniciar el aprendizaje debido a su legibilidad y facilidad de uso. Una de las ventajas principales de este lenguaje es que posible escribir programas en menos líneas de código que un programa equivalente de C / C ++ o Java.(Kelly, 2019).

2.5.1.3.Tensor Flow

TensorFlow es una plataforma de software de código abierto desarrollada por Google para la creación de modelos de aprendizaje automático y redes neuronales. También utiliza gráficos de flujo de datos para representar el modelo y ejecutar cálculos numéricos intensivos mediante CPUs o GPUs. TensorFlow es utilizado ampliamente en aplicaciones de aprendizaje automático, como reconocimiento de voz e imágenes como es la visión artificial. La plataforma también incluye una variedad de herramientas para ayudar en la obtención, procesamiento, entrenamiento y despliegue de modelos de aprendizaje automático.(Géron, 2022)

Actualmente, la biblioteca de aprendizaje profundo más utilizada de manera general es Tensor Flow de Google. Este desarrollo de Google utiliza el aprendizaje automático en todos sus

productos que permitan optimizar el motor de búsqueda, la traducción, los subtítulos de imágenes en sistemas de predicción o análisis.(Rungta, 2018)

Para ejemplificar de mejor manera es la identificación mediante Google para experimentar una búsqueda más eficiente y minuciosa a través de IA. Es cuestión de digitar una palabra clave en la barra de búsqueda, el motor Google proporciona una recomendación sobre cuál podría ser la siguiente palabra. TensorFlow opera mediante tres parámetros principales es decir cuenta con una arquitectura de:

- Preprocesamiento de los datos
- Generar el modelo
- Entrenar y estimar el modelo

La plataforma TensorFlow se denomina porque toma la entrada como una matriz multidimensional, también conocida como tensores. Puede construir un tipo de diagrama de flujo que permita realizar operaciones (llamado gráfico) que inicia en la entrada adquisición de datos. Los datos inician en un extremo para luego fluir a través de este sistema de múltiples operaciones y finalmente se obtiene por el otro extremo como salida.(Rungta, 2018)

2.5.1.4.Teachable Machine

La herramienta desarrollada por Google para aplicaciones en IA es que permite al usuario mediante el uso de una plataforma web crear aprendizaje automático de manera rápida, fácil y accesible para que se empleado en proyectos o aplicaciones de entrenamiento y se denomina Teachable Machine. Los usuarios pueden enseñar a la máquina a reconocer objetos, sonidos y gestos mediante la captura y el etiquetado de datos en tiempo real utilizando la cámara web y el micrófono de su computadora. La herramienta se basa en TensorFlow.js, una biblioteca de JavaScript para el aprendizaje automático en el navegador. Teachable Machine se puede utilizar

para una variedad de aplicaciones, desde la automatización de tareas hasta la creación de experiencias interactivas en línea. Web a la que se puede acceder a través del sitio https://teachablemachine.withgoogle.com. Practicar el aprendizaje automático en la web oficial es muy fácil y solo requiere tres pasos en crear un proyecto, como primer paso es recopilar datos, para luego proceder a estudiar el modelo y el finalmente es exportar el proceso o los resultados del aprendizaje automático que usará para implementar en la aplicación en este caso la visión por computadora o visión artificial.(Agustian et al., 2021). En la Figura 3 se puede identificar el proceso de iniciar un proyecto en la web oficial de Teachable Machine.

Figura 4

Interfaz Gráfica de Teachable Machine desde su web oficial

*Nota***:** Mediante la página oficial de Teachable machine se puede acceder a realizar entrenamientos de sistemas para visión artificial. Tomado de (*Teachable Machine*, n.d.)

En el sitio web de Teachable Machine de Google cuenta con 3 opciones que pueden ser seleccionado para crear aprendizaje automático, es necesario establecer un método para iniciar el entrenamiento, el modo de clasificación de imágenes, la clasificación de voz y la clasificación de

poses. La clasificación de imágenes y poses es una clasificación diferente, donde la imagen de clasificación solo se centra en una imagen y la clasificación de poses se centra en poses humanas. En la clasificación utilizando la máquina Tachable de Google puede muestrear un dato existente previamente guardado o tomar fotos directamente en la cámara web o grabar sonidos mediante el micrófono o archivos de audio previamente generados y almacenados en algún dispositivo o repositorio.(Agustian et al., 2021)

2.5.2. Hardware aplicado a la visión artificial

La visión artificial es la disciplina que tiene como objetivo interpretar la estructura de las imágenes para luego utilizarlo en diferentes procesos. La CV es un campo de IA cuyo propósito es permitir que una máquina entienda lo que ve cuando lo conectamos a una o más cámaras. Con la aparición de procesadores más eficientes, se vuelve práctico incorporar algoritmos de visión artificial en sistemas integrados, lo que permite analizar sus entornos a través de la adquisición de imagen desde una fotografía o a través de un video, la misma ser almacenada en su memoria. Con el avance tecnológico, en temas de la microelectrónica traen nuevas herramientas y dispositivos integrados que permiten que los sistemas de visión artificial diseñen de manera eficiente y capaz de cumplir con tareas específicas asignados por el usuario.(Messaoud et al., 2022)

Existen en el mercado tecnológico diversos elementos que posibilitan ejecutar un sistema de visión artificial, cuentan con características específicamente relacionados a procesamiento de información, juntamente con sensores o cámaras de alta resolución para la toma de datos en tiempo real de tal forma que permita estructurar adecuadamente un sistema de visión artificial. A continuación, se muestra ejemplos de estos equipos.

2.5.2.1.Raspberry PI

Según Tim Cox (2016) en su libro Raspberry Pi for Python Programmers Cookbook dice que: “La Raspberry Pi” es una computadora de placa única creada por la Fundación Raspberry Pi, una organización benéfica formada con el propósito principal de reintroducir habilidades informáticas de bajo nivel para niños en el Reino Unido. El objetivo era reavivar la revolución de los microordenadores de la década de 1980, que produjo toda una generación de programadores cualificados.(Cox, 2016)

Raspberry Pi es una familia de SBC organizado por la Fundación Raspberry Pi, han desarrollado varios modelos de placas que cuentan con diferentes características, hoy en día es posible armar una Pc personal utilizando estas placas. Los SBC se utilizan como demostradores de tecnología (prototipos), computadoras educativas y sistemas integrados. Hay un aumento reciente en la popularidad de los SBC debido a los avances en el proceso de fabricación y las tecnologías de fabricación. Algunas familias prominentes de SBC son Raspberry Pi, Banana Pro, BeagleBoards y Orange Pi. Raspberry Pi es la familia más popular de computadoras de placa única disponible en el mercado, y es una de las computadoras más vendidas en el mundo. (Pajankar, 2021)

2.5.2.2.Raspberry OS

El sistema operativo Raspberry Pi es un derivado de una popular distribución de Linux conocida como Debian. Es proporcionado oficialmente por la Fundación Raspberry Pi, y es el sistema operativo más recomendado para la familia RPi de SBC. Está totalmente optimizado para los modelos de placa RPi, y todos los modelos de placa son compatibles con ella. Anteriormente, era conocido como el sistema operativo Raspbian, por su creador por Peter Green y Mike Thompson.(Pajankar, 2021)

2.5.2.3.Cámara PI

El módulo de cámara Raspberry Pi v2 es un componente diseñado específicamente para la Raspberry Pi, está compuesto por un sensor de imagen Sony IMX219 de alta calidad de 8 megapíxeles y una lente de enfoque fijo. Este módulo tiene la capacidad de capturar imágenes estáticas de alta resolución de 3280 x 2464 píxeles, y de grabar video en 1080p30, 720p60 y 640x480p60/90. Este módulo se conecta a la Raspberry Pi mediante a través de enchufes en la parte superior de la placa. La cámara también cuenta con un sensor de imagen Sony IMX219 de alta calidad con una resolución nativa de 8 megapíxeles y una lente de enfoque fijo incorporada, lo que permite capturar imágenes fijas de alta calidad y grabar videos en diferentes resoluciones.(*Raspberry Pi Camera Module V2*, n.d.)

2.6.Inteligencia Artificial

La IA se define como la capacidad de los ordenadores para realizar tareas que normalmente requieren ejecutarse mediante inteligencia humana. Sin embargo, se establece una definición de manera general interpretando que la IA permite a las máquinas utilizar e interpretar algoritmos para aprender sobre los datos obtenidos y tomar decisiones similares a las de un ser humano. A diferencia de las personas, los dispositivos que contengan IA pueden analizar grandes cantidades de información sin periodos de descanso y con una proporción de errores significativamente menor que sus homólogos humanos al realizar tareas similares.(ROUHIAINEN, 2018)

Las áreas de aplicación de la IA son en casi todas las áreas o ambientes a continuación se muestran las principales técnicas de la IA aplicados en diferentes medios de la industria y a la par se trata de mejorar su estabilidad por lo que se están desarrollándose nuevas prestaciones de la IA y su constante investigación que crece a paso acelerado en la actualidad:

- **Automatización de procesos empresariales**: La IA es posible aplicar para automatizar tareas y procesos repetitivos, como la clasificación de correos electrónicos, la gestión de inventarios, la monitorización de calidad, entre otros.
- **Industria manufacturera**: Mediante la IA se realiza procesos para mejorar la eficiencia y la calidad en la producción, también aplicado en realizar tareas de control de calidad.
- **Sector salud**: En la industria de la medicina la IA cumple un papel fundamental en en la interpretación de imágenes médicas, en la predicción de enfermedades y en la personalización de tratamientos médicos.
- **Sector financiero**: En el sector financiero la IA es aplicado en el análisis de riesgo crediticio, la detección de fraudes, la automatización de procesos de crédito, y la predicción de tendencias financieras.
- **Marketing y publicidad**: Para generar contenido en diferentes plataformas la IA es utiliza para la segmentación de audiencias, la personalización de mensajes publicitarios, el análisis de sentimiento en redes sociales, y la optimización de la publicidad en línea.
- **Automoción:** En sector automovilístico aplicar la IA es fundamental porque se están desarrollando vehículos autónomos capaces de trasladarse de un puno a otro de manera autónoma, de igual forma en el análisis de datos de sensores, y en la mejora de la eficiencia de los vehículos.
- **Agricultura**: Para la industria del agro el desarrollo tecnológico es una de las herramientas que permiten a mejorar o realizar controles en la calidad de sus productos de tal manera que la IA se utiliza para la detección temprana de

enfermedades en plantas, en la predicción de rendimientos de cultivos, y en la optimización de la gestión de recursos agrícolas.

2.7.Machine Learning

El aprendizaje automático es un subcampo de la IA, en la actualidad se hacen estudios e investigaciones de aplicabilidad en diferentes áreas ya que el aprendizaje automático tiene un rápido crecimiento permitiendo a las maquinas realizar tareas específicas, la principal dificultad está en el diseño y análisis de algoritmos que permitan implementar estos algoritmos en diferentes campos. Si bien todavía es una disciplina joven, con mucho más por descubrir de lo que se sabe actualmente, hoy en día el aprendizaje automático se puede utilizar para enseñar a las computadoras a realizar una amplia gama de tareas útiles.(Watt et al., 2016)

Mediante el aprendizaje automático es posible realizar tareas específicas, estas tareas incluye como la detección automática de objetos en imágenes (un componente crucial de los automóviles asistidos por el conductor y autónomos), el reconocimiento de voz (que impulsa la tecnología de comando de voz), el descubrimiento de conocimiento en las ciencias médicas (utilizado para mejorar nuestra comprensión de enfermedades complejas) y el análisis predictivo (aprovechado para ventas y pronósticos económicos).(Watt et al., 2016)

De tal manera que existen muchos subcampos y diferentes aplicaciones dentro de la IA, el aprendizaje automático (ML) en la actualidad se ha vuelto extremadamente popular como una herramienta práctica para muchos campos de aplicaciones infundidas con IA disponibles en la actualidad. ML es una rama de la informática donde a las computadoras se les enseña a procesar información y tomar decisiones al dar acceso a los datos de los cuales las computadoras aprenden.(Watt et al., 2016)

La ejecución de tareas más comunes de ML incluye clasificación, regresión, recomendaciones, clasificación y agrupación de datos. Por lo tanto, la IA es un concepto más amplio que barca al subcampo como es la ML, mientas que ML es un área de investigación dentro de la IA en torno a una ideología de que las máquinas pueden aprender por sí mismas a realizar cualquier actividad una vez que se le da acceso a la información o datos para su uso.(Salvaris et al., 2018).

Una computadora es capaz de aprender a realizar procesos para modelar la información obtenida en la entrada y el resultado será datos que la maquina procesó y está tratando de predecir; El tipo más común de ML es el aprendizaje automático supervisado, donde el modelo tiene etiquetas que se supone que representan la verdad básica contra la cual aprender. El proceso de aprendizaje de la computadora de los parámetros dentro del modelo a menudo se llama entrenamiento.(Salvaris et al., 2018).

2.7.1. Modelos de aprendizaje automático (ML)

En el aprendizaje automático se tiene procesos que se llaman métodos, estos comprenden en que cada etapa de aprendizaje automático cuenta con subetapas donde los datos obtenidos los identifican mediante etiquetas para luego aplicarlo o ejecutarlo en una computadora a través de un programa para definir las características de los datos obtenidos y finalmente también se puede agregar métodos si previo entrenamiento. A continuación, se muestra a cada uno de estos métodos: aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo.

2.7.1.1.Aprendizaje supervisado

En el aprendizaje supervisado tiene como objetivo principal establecer una relación entre los datos de entradas con los datos de salida, en el aprendizaje supervisado, se procede a establecer etiquetas de datos obtenidos para el algoritmo en su fase inicial, la cual es el entrenamiento, este

proceso es fundamental para la supervisión en etapa de entrenamiento con lo cual es obtener un sistema que en su mayoría no contenga errores. Los datos resultantes generalmente son etiquetados de manera manual y sirven como verdad básica para el algoritmo. El objetivo del algoritmo es aprender una regla general que asigna desde las entradas hacia las salidas. En el aprendizaje automático, la verdad básica se refiere a los datos que se supone que son verdaderos. (Rees & Ryder, 2022)

A continuación, se detalla algunos de los algoritmos empleados en el aprendizaje supervisado más relevantes:

- k-Vecinos más cercanos
- Regresión lineal
- Regresión logística
- Máquinas de vectores de soporte (SVM)
- Árboles de decisión y bosques aleatorios
- Redes neuronales

2.7.1.2.Aprendizaje No supervisado

En el aprendizaje no supervisado, no se incluye información de etiquetado de datos en el algoritmo de aprendizaje. Este tipo de algoritmos trabaja únicamente con datos de entrada y su objetivo principal es identificar correlaciones o patrones entre cada uno de los datos ingresados, lo que a menudo se llama "Descubrimiento de conocimiento". La principal diferencia con el aprendizaje supervisado radica en que no se especifica qué tipo de correlaciones se buscan o se predicen, lo que significa que no hay una métrica obvia para medir el error cometido por el algoritmo.(De & Ordoba, 2019), a continuación, se detalla los principales algoritmos utilizado en el aprendizaje no supervisado:

- K-Means
- Mixturas de Gaussianas
- Análisis de Componentes Principales (PCA)

2.7.1.3.Aprendizaje por Refuerzo

El aprendizaje por refuerzo es una técnica de aprendizaje automático en la que un agente interactúa con un entorno dinámico y recibe retroalimentación en forma de recompensas o castigos por cada acción que realiza. El objetivo del agente es aprender a tomar las mejores decisiones para maximizar su recompensa a largo plazo. A diferencia del aprendizaje supervisado y no supervisado, el aprendizaje por refuerzo no tiene un conjunto de datos etiquetados y el agente debe explorar y aprender de su entorno a través de ensayo y error. Este tipo de aprendizaje se utiliza comúnmente en aplicaciones como robótica, juegos y automatización industrial.(gaitan & ampudia, n.d.)

Este método de aprendizaje automático se desarrolla en un entorno dinámico y se utiliza tanto datos etiquetados como no etiquetados. El sistema interactúa con su entorno y recibe retroalimentación positiva o negativa en función de sus acciones, lo que le permite mejorar y desarrollar mejores características y clasificaciones. Por lo tanto, la computadora aprende sin recibir instrucciones explícitas. Estas son las principales categorías del aprendizaje automático y se subclasifican en diferentes tipos de aprendizaje automatizado según la meta que se quiera lograr.(Choy et al., 2018)

2.8.Deep Learning

El aprendizaje profundo, también conocido como Deep Learning, es una técnica de aprendizaje automático que se basa en la construcción de algoritmos que imitan el funcionamiento del cerebro humano para procesar información y aprender a partir de ella. Esta técnica se utiliza principalmente para tareas que involucran grandes conjuntos de datos, como el reconocimiento de voz, la clasificación de imágenes y la traducción automática, entre otros. El aprendizaje profundo se basa en la construcción de redes neuronales artificiales compuestas por múltiples capas que procesan la información de manera jerárquica y aprenden a partir de la retroalimentación que reciben.(ROUHIAINEN, 2018)

Según Brownlee Jason (2017) en su libro: Deep Learning for Natural Language Processing: "El aprendizaje profundo" es un subcampo del aprendizaje automático relacionado con algoritmos inspirados en la estructura y función del cerebro llamadas redes neuronales artificiales. En la actualidad el aprendizaje profundo es aplicado en clasificación de imágenes, detección de voz y entre otros.(Brownlee Disclaimer, 2017)

2.8.1. Métodos de Deep Learning

Los métodos de aprendizaje profundo son técnicas de aprendizaje automático que involucran la utilización de redes neuronales artificiales con múltiples capas de procesamiento. A diferencia de los métodos tradicionales de aprendizaje automático, que utilizan una o dos capas de procesamiento para extraer características principales de los datos de entrada, los métodos de aprendizaje profundo utilizan múltiples capas de procesamiento, lo que les permite aprender representaciones de los datos de entrada de manera jerárquica y más compleja detallando estructura y formas de datos.

Los métodos desarrollados para el aprendizaje profundo han demostrado ser muy efectivos en una amplia gama de aplicaciones, incluyendo reconocimiento de voz, reconocimiento de

objetos en imágenes, procesamiento de lenguaje natural, y más. Los métodos de aprendizaje profundo han impulsado muchos avances en la inteligencia artificial en los últimos años y se han convertido en una herramienta importante para resolver problemas complejos de la vida real.

2.8.1.1.Red neuronal convolucional (CNN)

Las CNN se aplican principalmente para tareas relacionadas con la visión por computadora y el reconocimiento de voz. Son capaces de abordar tareas que involucran conjuntos de datos con relaciones espaciales, donde las columnas y filas no son intercambiables (por ejemplo, datos de imagen). Su arquitectura de red comprende una serie de etapas que permiten el aprendizaje jerárquico de características según lo determinado por la tarea de modelado respectiva. (Janiesch et al., n.d.)

2.8.1.2.Red neuronal recurrente (RNN)

Una Red Neuronal Recurrente (RNN, por sus siglas en inglés) es un tipo de red neuronal artificial que se utiliza en el aprendizaje profundo y que está diseñada para procesar datos secuenciales. A diferencia de las redes neuronales convencionales, las RNN tienen conexiones que forman un bucle, lo que les permite tener memoria y utilizar la información de entrada anterior para influir en la salida actual. De esta manera, las RNN pueden modelar patrones en datos secuenciales como texto, audio, video, entre otros. Las RNN se han utilizado en una amplia gama de aplicaciones, desde reconocimiento de voz y escritura a mano, hasta traducción automática y generación de texto y música.

2.8.1.3.Representación distribuida

Las representaciones distribuidas desempeñan un papel esencial en el aprendizaje de características y el modelado del lenguaje en tareas de PNL, donde las entidades lingüísticas como palabras, frases y oraciones se proyectan en representaciones numéricas dentro de un espacio

semántico unificado en forma de incrustaciones. Las incrustaciones de palabras, por ejemplo, codifican palabras discretas en vectores de características densas con baja dimensionalidad. Por lo tanto, a diferencia de los modelos clásicos de representación de texto, como las codificaciones de un solo calor y la bolsa de palabras (BoW), las incrustaciones de palabras superan el problema de las codificaciones dispersas al tiempo que preservan las relaciones semánticas entre las palabras. (Lecun et al., 2015)

2.8.1.4.Red neuronal generativa antagónica (GAN)

La Red Neuronal Generativa Antagónica (GAN) es un tipo de arquitectura de redes neuronales artificiales que se utiliza para el aprendizaje profundo no supervisado. Se compone de dos redes neuronales, el generador y el discriminador, que compiten entre sí en un proceso de entrenamiento para mejorar el rendimiento de la red.

El generador toma una muestra aleatoria de ruido como entrada y genera una imagen sintética a partir de ella. El discriminador, por otro lado, toma una imagen como entrada y decide si es real o sintética. El objetivo del generador es engañar al discriminador para que clasifique las imágenes sintéticas como reales, mientras que el discriminador trata de identificar correctamente las imágenes reales y sintéticas.

A medida que la GAN se entrena, el generador aprende a producir imágenes cada vez más realistas, mientras que el discriminador aprende a ser más efectivo en la identificación de imágenes reales y sintéticas. La GAN ha demostrado ser muy efectiva en la generación de imágenes sintéticas de alta calidad, lo que la hace útil en aplicaciones como la creación de arte, la síntesis de imágenes médicas y la animación.(Yepes et al., 2020)

Capítulo III

Diseño del Sistema

En este capítulo se procede a identificar las estrategias para el proceso de diseño del prototipo la cual es un sistema de clasificación del grano de chocho (Lupinus Multabilis Sweet), mediante el uso de visión artificial, en este se detalla los requerimientos del sistema, descripción en bloques el funcionamiento del sistema, de igual manera la identificación de la arquitectura del sistema mediante el uso de diagramas de bloques. Al tener definido los parámetros correspondientes al diseño del sistema se procede a estructurar el desarrollo del proyecto mediante la metodología Interactiva

3. Metodología

El modelo interactivo es una metodología de desarrollo de software que se centra en la participación de los usuarios y las partes interesadas a lo largo de todo el proceso. A diferencia de los enfoques lineales y secuenciales, el modelo interactivo promueve una interacción constante entre los desarrolladores y los usuarios, permitiendo una retroalimentación temprana y continua.

En el modelo interactivo, el desarrollo del software o del hardware es dividido en ciclos o iteraciones, donde cada ciclo incluye etapas de diseño, implementación, prueba y evaluación. A medida que se completa un ciclo, el software o el hardware resultante se presenta y se recopila la retroalimentación de los usuarios, lo que ayuda a refinar y mejorar el producto.

La metodología iterativa es un enfoque de desarrollo de software que se basa en la realización de ciclos repetitivos de trabajo, conocidos como iteraciones, para desarrollar un producto o sistema. A diferencia de los enfoques lineales y secuenciales, la metodología iterativa permite la entrega gradual de funcionalidades y la obtención de retroalimentación temprana y

frecuente de los usuarios y las partes interesadas. A continuación, en la figura 5 se presenta la estructura de la modelo iterativa.

Figura 5

Estructura del proceso del modelo Iterativa

Nota: Estructura del modelo metodológico-iterativa. Tomado de (Javier et al., n.d.)

En metodología iterativa, cada iteración comprende las etapas de análisis, diseño, implementación y prueba. En lugar de intentar completar todo el proyecto de una sola vez, se divide en incrementos más pequeños y manejables. Cada iteración tiene un objetivo específico y se enfoca en desarrollar y mejorar un conjunto de funcionalidades.

3.1.Modelo Iterativa: Primera iteración

La principal característica de la metodología iterativa es que permite la adaptabilidad y flexibilidad a medida que se obtiene más información y se recibe retroalimentación del usuario.

Esto significa que los requisitos y objetivos del proyecto pueden ajustarse y refinarse en cada iteración, lo que lleva a un producto final más sólido y satisfactorio.

3.1.1. Requerimientos del sistema: Primera iteración

Según los requerimientos establecidos por el estándar internacional ISO/IEC / IEEE 29148: 2011 que se enfoca directamente en el desarrollo de software menciona que es fundamental integrar los siguientes ámbitos: Requerimiento de stakeholders, Requerimiento de Sistema y Requerimiento de Arquitectura.

3.1.1.1.Nomenclatura y requerimientos de procesos

De acuerdo con los requerimientos proporcionados es necesario detallar a cada uno de estos los cuales son tres, en la tabla2 se muestra su nomenclatura y su correspondiente definición, con lo cual se pretende tener un orden en el desarrollo del proyecto.

Tabla 2

Nomenclaturas

Nomenclatura	Definición
StSR	Requerimientos de Stakeholders
SySR	Requerimientos de Sistema
SRSH	Requerimientos de Arquitectura.

Nota. En esta tabla2 se muestra la nomenclatura utilizada para etiquetar los requerimientos del sistema en desarrollo. Fuente: Autoría Propia.

De igual manera se describe el grado de aplicabilidad o la utilidad de los requerimientos en el proceso de desarrollo del sistema por lo que es importante especificar la priorización de cada uno de los estados o requerimientos, por lo general se lo cataloga de tres formas en la tabla3 se muestra estas asignaciones.

Tabla 3

Categorías (Stakeholders)

Categoría	Descripción
Baja	Son el tipo de requerimientos que es posible aplicar al sistema de manera opcional y no dependerá para el funcionamiento del sistema
Media	Este requerimiento de implementación es opcional, sin embargo, si es que el sistema requiere omitirlo es posible
Alta	Para este tipo de requerimientos es obligatorio aplicarlo en el sistema ya que es una parte principal en el desarrollo del sistema.

Nota. A continuación, en la tabla3 se especifica el grado o categoría del requerimiento en su utilización transcurso el desarrollo. Fuente: Autoría propia

3.1.1.2.Requerimientos de Stakeholders

Los requisitos del sistema se derivan de la observación directa de las necesidades de los usuarios y la investigación sobre métodos y sistemas que permitan determinar la calidad del grano de chocho post cosecha. Es por lo que tiene un valor primordial en que estos requisitos sean determinados y analizados por diversos actores involucrados para el desarrollo del sistema, el cual se muestra en la Tabla4, de esta manera garantizar un desarrollo y funcionamiento correcto del sistema. En el proceso de desarrollo del sistema mediante iteraciones, existe la posibilidad de mejorar o eliminar aquellos requisitos agregados en el proceso de gestión del sistema.

Tabla 4

Requerimientos Stakeholders

Requerimientos de Stakeholders

Usuarios directos	Familia Farinango
Administradores	Mario David Farinango Túquerres
Director y asesor del proyecto	MSc. Edgar Maya

Nota. En la tabla4 se especifica las personas directamente interesadas en el desarrollo del proyecto (sistema embebido). Fuente: Autoría propia.

En el desarrollo del sistema es necesario establecer requerimientos que son importantes para el equipo el cual va a tener acceso a este sistema, en este comprende de técnico y como también del usuario final. Por lo cual en la siguiente tabla5 se describe los requerimientos, de acuerdo con su grado de aplicabilidad.

Tabla 5

Requerimientos de funcionalidad

REQUERIMIENTOS DE FUNCIONALIDAD				
N	**Requerimientos**	**Prioridad**		
#	**Requerimientos Operacionales**	**Alta**	**Media**	**Baja**
StSR1	El sistema embebido debe contar con una cámara de buena resolución	X		
StSR2	El sistema embebido debe ser fácil de utilizar	X		
StSR3	El sistema está enfocado a los agricultores de la comunidad de el abra.	X		
StSR4	El sistema embebido no debe ocupar mucho espacio físico.		X	
StSR5	El sistema embebido debe tener conexión a internet para validar con el sistema entrenado		X	

StSR6	El sistema embebido debe operar de forma autónoma en el proceso de clasificación del grano	X

Nota. En la tabla 5 se establece los requerimientos de operación del sistema. Fuente: Autoría propia.

3.1.1.3.Requerimientos de funcionalidad del sistema

De acuerdo con el procedimiento en el desarrollo de sistema es necesario definir los requerimientos del sistema, los cuales son un conjunto de funcionalidades que estructuran el sistema propuesto esto se hace mención en la tabla6. Los requerimientos del sistema pueden estar estructurados de la siguiente manera: Requerimientos de uso, Interfaces, estados físicos.

Tabla 6

Requerimientos de Sistema

REQUERIMIENTOS DE SISTEMA				
N	**Requerimientos**	**Prioridad**		
#	**Requerimientos de uso**	**Alta**	**Media**	**Baja**
SySR1	EL sistema embebido debe contar con una fuente de alimentación	X		
SySR2	El sistema debe tener facilidad de uso por el usuario	X		
SySR3	El sistema debe ser capaz de procesar datos rápidamente.	X		
SySR4	Botón de encendido y apagado del sistema	X		
Requerimientos de Interfaces				
SySR5	El equipo debe contar con puerto para cámara	X		

SySR6	El equipo debe contar con un sistema de ventilación	X		
SySR7	Sistema debe tener puerto o interfaz para acceso a internet		X	
	Requerimientos Físicos			
SySR8	El sistema debe ser fácil de transportar		X	
SySR9	La Cámara debe tener un buen ángulo de apertura	X		
SySR10	Las Imágenes obtenidas del grano de chocho deben ser de buena calidad	X		

Nota. La tabla 6 muestra los requerimientos principales de uso como: uso, interfaces y físicos del sistema embebido. Fuente: Autoría propia

3.1.1.4.Requerimientos de arquitectura

En los requerimientos de arquitectura (SRSH) se define a aquellos requerimientos que contemplan la estructura del sistema, tales como el software, hardware y la fuente eléctrica, en la tabla se identifica los parámetros del requerimiento de la arquitectura, mediante este facilita la elección del hardware y de igual forma el software.

Tabla 7

Requerimientos de arquitectura

Requerimientos de arquitectura				
N	**Requerimientos**	**Prioridad**		
#	**Requerimientos de Diseño**	**Alta**	**Media**	**Baja**
SRSH1	El sistema debe contar con una pantalla donde permita visualizar la información		X	
SRSH2	La cámara debe enfocar de forma precisa cada grano de chocho	X		
SRSH3	Sistema debe tener acceso a internet	X		

	Requerimientos de Hardware	
SRSH4	El sistema Embebido debe tener un puerto para la conexión de una cámara	X
SRSH5	La cámara debe tener compatibilidad con el sistema operativo de la placa.	X
SRSH6	El procesador del equipo embebido debe ser capaz de procesar información de alta capacidad	X
SRSH7	El sistema embebido debe tener un sistema de energía para su alimentación.	X
	Requerimientos de software	
SRSH8	Lenguaje de programación open Source	X
SRSH9	Sistema operativo de código abierto	X
SRSH10	Sistema operativo compatible con el lenguaje de programación	X
SRSH11	Lenguaje de programación que admita adjuntar la cámara de forma lógica	X

Nota. En la tabla 7 se identifica la información sobre requerimientos de hardware, software y como también del diseño del sistema embebido. Fuente: Autoría propia

3.1.2. Selección de Hardware y software

Para la estructuración del sistema embebido es fundamental la selección de hardware y software de acuerdo con un análisis de requerimientos ya establecidos con anterioridad, para la selección es fundamental identificar el equipo con características de procesamiento de datos a gran cantidad de igual forma el software adecuado para que permita interactuar óptimamente entre el sistema embebido. A continuación, se ha estructurado una tabla que facilite la identificación de hardware como software con valores ya establecidos en los requerimientos y que estos se cumplan de acuerdo con su uso.

Tabla 8

Descripción de valores

Descripción de Valores	
Si cumple	1
No cumple	0

Nota. Describe los valores a agregar en el proceso de selección. Fuente: Autoría propia.

3.1.2.1.Elección de Hardware

En la elección de hardware se realiza una comparativa de los dispositivos que permiten el correcto funcionamiento del sistema en funciones de adquisición y procesamiento de datos. El hardware principal es el encargado de toda la gestión del sistema embebido por lo cual sus características deben cumplir con los requerimientos previamente establecidos.

Tabla 9

Selección de la placa SBC

PLACAS SBC							
Hardware (Nombre del dispositivo)	**Requerimientos**						**Valoración**
	StSR2	**StSR4**	**SySR4**	**SySR5**	**SRSH2**	**SRSH5**	Total
Raspberry Pi Zero	0	1	1	1	1	1	5
Raspberry Pi Compute	1	1	1	1	1	0	5
Raspberry Pi 4 modelo B	1	1	1	1	1	0	5
Raspberry Pi 5	1	1	1	1	1	1	6
Arduino Mega	1	1	0	1	1	1	5
Arduino Uno	0	1	0	1	1	1	4

1 cumple	
0 no Cumple	
Selección:	Raspberry Pi 5

Nota. La tabla9 muestra las placas SBC principales para la implementación del sistema embebido, se clasifica de acuerdo con los requerimientos establecidos. Fuente: Autoría propia

Como se evidencia en la tabla anterior, se realizó un análisis de las características correspondiente a los requerimientos de Stakeholders para cada una de las placas SBC si cumplen o no para la integración en el sistema embebido, por lo tanto, se ha realizado una valoración de cada parámetro para definir que placa es óptima para este sistema.

Para complementar la valoración de la tabla anterior de procedió a especificar las características más importantes de la placa SBC con mayor puntuación, es por lo tanto en la tabla se visualiza las principales características de la placa.

Tabla 10

Especificaciones técnicas del Raspberry Pi5

Parámetros	Descripción
Procesador	• 4x Cortex-A76 2.4GHz, con Crypto Extensions. • VideoCore VII, OpenGL ES 3.1, Vulkan 1.2
Memoria RAM	• LPDDR4X-4267, 2GB, 4GB y 8GB disponibles
Puertos	• 2x USB 3.0, 2x USB 2.0 • Dual 4Kp60 HDMI con soporte HDR
Alimentación	• 5V/5A DC por USB-C, con Power Delivery
Conectividad	• Dual-band 802.11ac Wi-Fi, Bluetooth 5.0/BLE

	• Gigabit, con PoE+ support
Almacenamiento	• Ranura para tarjetas microSD, compatible con el modo SDR104
Características Adicionales	• RTC con batería externa, Botón de encendido

Nota. La tabla 10 muestra las características principales de la placa SBC Raspberry pi5. Fuente: Autoría propia

3.1.3. Sensor (Cámara)

El dispositivo principal del sistema embebido es el sensor o cámara ya que este es el encargado de capturar las imágenes que corresponden al estado del grano de chocho, el sensor debe ser compatible con la placa SBC para su correcto funcionamiento, en la tabla se realiza una comparativa de sensores que son compatibles con la placa SBC, en este caso Raspberry PI5 detallado anteriormente en la tabla.

Tabla 11

Selección de sensor(cámara)

Selección del Sensor					
Hardware (Nombre del dispositivo)	**Requerimientos**				**Valoración**
	SRSH2	**SRSH5**	**SySR9**	**StSR1**	**Total**
Raspberry Pi Camera v2 (Adafruit)	0	1	1	1	3
Arducam Mini Raspberry Pi	0	1	1	1	3
HQ Camera	1	0	1	1	3
Raspberry Pi High-Quality	0	1	1	0	2
Camera (Waveshare)					
Raspberry Pi HQ de alta	1	1	0	1	3
calidad					

Cámara HQ con dos lentes distintas	1	1	1	1	4
1 cumple 0 no Cumple Selección:	Cámara Raspberry Pi HQ de alta calidad				

Nota. La tabla11 Selección de sensor para la integración del sistema. Fuente: Autoría propia

Con la referencia de la tabla anterior tabla se procedió a detallar las principales características del sensor, encargado de la adquisición de imágenes en el sistema embebido. En la siguiente tabla se muestra las funcionalidades más importantes de este dispositivo.

Tabla 12

Especificación de las principales características del sensor

Parámetros	Descripción
Sensor de cámara	Sony IMX477R con retroiluminación
Formato Óptico	1/2,3” (7,9 mm) o más grande
Resolución	12.3 megapíxeles
Tamaño de píxel	1.55µm × 1.55µm
Tipo de enfoque	Ajustable (12.5 mm–22.4 mm)
Longitud del cable plano	200mm
Salida	RAW12/10/8, COMP8

Nota. La tabla 12 Descripción de las características del sensor(cámara). Fuente: Autoría propia

3.1.3.1.Elección de Software

Para la elección del software es necesario comprender dos factores muy importantes: el sistema operativo debe ser compatible con la placa SBC y el lenguaje de programación desarrollado para adquisición y procesamiento de datos del sistema embebido, por lo cual los dos parámetros deben ser compatibles entre sí.

3.1.3.1.1. Selección de Sistema Operativo

En la elección del sistema operativo es fundamental tener en cuenta tanto su arquitectura como es su procesamiento en una placa SBC. A continuación, se muestra en la tabla una comparativa entre distintos sistemas operativos compatibles con la placa sbc, en la tabla se muestra los distintos tipos de sistema operativo de acuerdo con los requerimientos de arquitectura.

Tabla 13

Elección de Sistema operativo

Selección del sistema operativo					
Hardware	**Requerimientos**				**Valoración**
	StSR2	**SySR2**	**SRSH8**	**SRSH10**	**Total**
Raspberry PI OS	1	1	1	1	4
Linux-Ubuntu	1	1	0	1	3
Windows	1	0	1	1	3
Arch Linux ARM	1	0	1	1	3
1 cumple					
0 no Cumple					
Selección:	Raspberry PI OS				

Nota. La tabla 13. Selección de sistema operativo para la placa. Fuente: Autoría propia

3.1.3.1.2. Lenguaje de programación

De acuerdo con los requerimientos establecidos en la sección anterior el lenguaje de programación debe tener la funcionalidad de operar los componentes del sistema embebido como también soportar modelos de entrenamiento de datos y compatibilidad con librerías requeridas para su funcionamiento, en la siguiente tabla se especifica diferentes lenguajes de programación y de esta manera determinar cuál es óptimo en los requerimientos del sistema embebido.

Tabla 14

Lenguaje de Programación

Selección del Lenguaje de Programación			
Software	**Requerimientos**		**Valoración**
	SRSH8	**SRSH8**	**Total**
Python	1	1	2
Java	0	1	1
C++	0	1	1
1 cumple			
0 no Cumple			
Selección: Python			

Nota. La tabla 14. Principales lenguajes de programación que soporta la placa SBC.

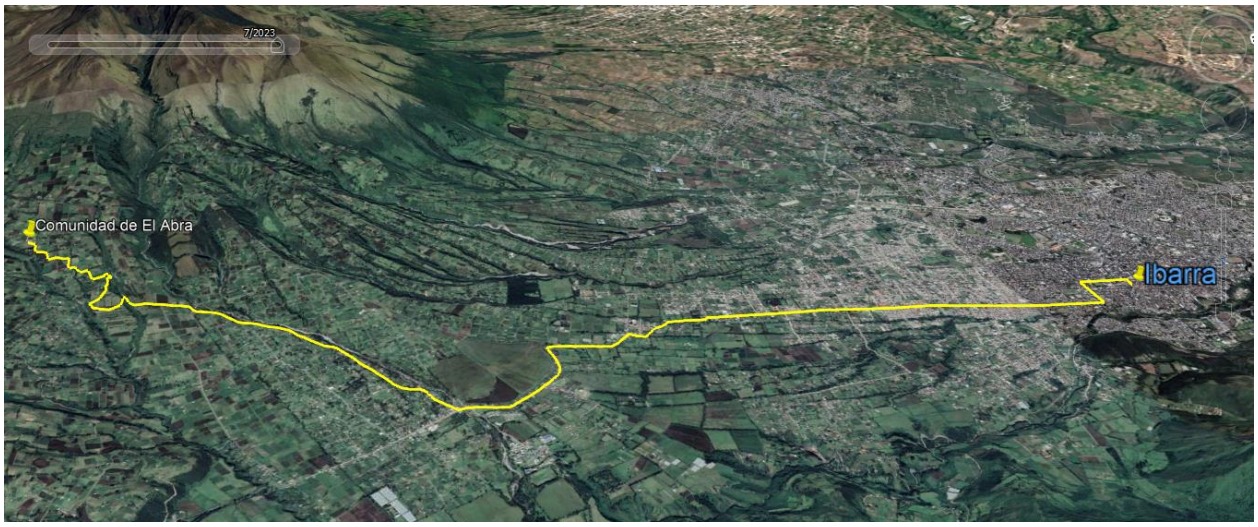
Fuente: Autoría propia

3.2.Situación actual: Primera iteración

El desarrollo del proyecto se enfoca en la clasificación del grano de chocho post cosecha aplicando técnicas de visión artificial, mediante este reducir el tiempo empleado en la clasificación o selección del grano de chocho, la selección del grano de chocho se lo realiza de forma manual por los agricultores de la comunidad de el abra. En la figura 2 se muestra la ubicación de la comunidad de el Abra.

Figura 6

Comunidad de el abra



Nota. Ubicación de la comunidad de EL Abra trazada en el aplicativo Google Earth. Fuente: Autoría propia

A lo largo de los años la tecnología ha tomado gran relevancia en sistematizar procesos que facilitan a realizar diferentes trabajos en cualquier área de la industria, una de esta es el sector agrícola en el cual, habido un constante avance y mejora, con el objetivo de aumentar la productividad de igual forma el rendimiento en la producción agrícolas como de la industria en general se han desarrollado técnicas o herramientas que sean aplicables en procesos agrícolas. Anteriormente, en el sector agrícola se caracterizaba por depender en gran medida del trabajo manual en temas de producción, recolección y clasificación de productos.

En la actualidad se está implementando un término sobre la aplicación de la agricultura 4.0, esta hace referencia a un nuevo dispositivo para llevar a cabo las estrategias tradicionales de la empresa agrícola con la ayuda de la tecnología progresiva que incluye la robótica agrícola, la computación en la nube y el internet de las cosas, entre otros. Esta tecnología se caracteriza por el

uso de la inteligencia artificial y la automatización de la recopilación de información aplicable para la toma de decisiones, el seguimiento regular de los cultivos y otras técnicas agrícolas tradicionales.(CEPAL, 2022)

El agricultor debe producir en cantidad para tener una igualdad en sus gastos de producción, los productos en la mayoría son distribuidos a revendedores que no pagan lo que corresponde al precio del producto, de igual forma no hay un ente regulador que vigile y controle estos precios, por lo que el agricultor vende sus productos en precios establecidos por terceros.

En sectores rurales los agricultores no disponen de herramientas tecnologías que ayuden a optimizar sus proceso o trabajos para producir los productos, una de estas es la clasificación de granos, cada grano debe ser clasificado y verificado su calidad, este proceso se lo realiza de forma manual lo cual demanda mucho tiempo al productor.

Para desarrollar el sistema propuesto, se ha considerado hacer uso de las cosechas de los agricultores de la comunidad específicamente del señor “José Cesar Farinango Farinango”, estas muestras serán del grano de Lupinus Multabilis Sweet “chocho’’ postcosecha (grano seco).

El prototipo de un sistema embebido para la clasificación del grano de chocho contará con dispositivo electrónico, sistema de visualización de información procesada como también con una o más cámaras para identificar la calidad del grano de chocho. El equipo de procesamiento debe tener la capacidad de identificar el estado físico del grano de chocho de acuerdo con el proceso de entrenamiento previo realizado.

En el análisis de la situación actual se procedió en la recopilación de información que permita la identificación de los principales requerimientos para el diseño del prototipo acorde a las necesidades y objetivos establecidos para el desarrollo de este proyecto, la gran parte de la información adquirida es por parte de los agricultores pertenecientes a la comunidad de el Abra.

Los moradores de la comunidad de el Abra en su mayoría se dedican a la agricultura, como fuente principal de ingresos económicos para su subsistencia por lo cual siembran diferentes tipos de productos agrícolas como son papas, cebada, trigo, maíz, haba, alverja y una de ella es el chocho, este grano tiene la particularidad del tiempo empleado en su maduración ya que el tiempo utilizado en la etapa de maduración son casi aproximadamente de 11 a 12 meses. Para tener una buena cosecha es necesario aplicar nutrientes a la planta en la etapa de crecimiento y también se aplica en la etapa de floración, de esta manera garantizar que la vaina donde se almacena el grano de chocho este en buenas condiciones, en la figura se muestra la etapa de floración de la planta de chocho y también en la figura se visualiza la formación de vainas del grano de chocho.

Figura 7

Florecimiento de la planta de chocho

Nota. Descripción grafica de la planta de chocho y de igual forma la muestra de las vainas del chocho. Fuente: Autoría propia

El proceso de cosecha del grano de chocho se lo realiza ya cuando la corteza de la vaina también conocida como la cascara del grano de chocho está completamente seca o a su vez cuando la planta este seca tallo y ramificaciones deben tener un color café. En la figura a) se visualiza el

estado de maduración del grano de chocho listo para su cosecha, b) se realiza el proceso de cosecha con una oz para luego almacenar en un contenedor en este caso un saco limpio y seco.

El proceso de cosecha del grano de chocho se lo puede realizar de dos formas, la primera forma es el corte de vaina por vaina desde la planta del chocho ya seco, este se lo almacena en costales o sacos para luego almacenarlo en un lugar fresco libre de humedad y la segunda forma es cortar todo el tallo y ramas de la planta de chocho ya seco y se lo procede a realizar un montículo de la planta cortada para luego proceder a la trilla. La Figura muestra como es el proceso de cosecha vaina por vaina del grano seco.

Figura 8

Proceso de cosecha del grano de chocho

Nota. Planta de chocho seco y proceso de recolección del grano de chocho (vaina de chocho). Fuente: Autoría propia

El proceso de trilla se lo realiza con una maquina industrial (máquina de tilla de granos) en este se coloca el grano en la tolva de la trilladora, la vaina seca se introduce cuidadosamente, pasa por la zaranda por donde el grano es separado de la cascara para finalmente recolectar el grano ya pelado en un recipiente seco

Figura 9

Trilla del grano seco de chocho

Nota. Proceso de trillado del grano de chocho, se ingresa por la tolva la vaina del chocho y es quitado de la cascara para posterior salir granos de chocho. Fuente: Autoría propia

Luego del proceso de la trilla el grano de chocho es almacenado en sacos o costales para luego ser clasificado, en el proceso de la trilla el grano se alacena sin ningún proceso de selección, es decir no es posible separar el grano en buen estado del grano que está en mal estado. La Figura se muestra el almacenado del grano después de la trilla.

Figura 10

Almacenaje del grano luego de la trilla

Nota. Grano de chocho recolectado y almacenado en un recipiente(costal). Fuente: Autoría propia

Para el proceso de clasificación del grano del chocho cada agricultor lo realiza de forma manual, el grano almacenado en sacos es puesto en una mesa o en alguna superficie plana que posibilite identificar y descartar el grano en mala condición, este habiente debe contar con buena iluminación para proceder a su clasificación, cada agricultor lo identifican de grano a grano, este proceso para una sola persona demanda mucho tiempo en la clasificación del grano de chocho en buen estado. En la figura se visualiza el proceso que se realiza en la clasificación del grano de chocho, en la cual se muestra cómo se separa el grano en mal estado, proceso que se lo realiza de forma manual y grano a grano.

Figura 11

Proceso de selección de grano de chocho

Nota. Clasificación del grano de chocho postcosecha. Fuente: Autoría propia

Para el agricultor el precio elevado de los productos es la principal oportunidad de comercializarlo en precios justos ya que muchos de los productos que procede a ser comercializado su valor económico se deprecia considerablemente, esto depende de la calidad o estado del producto, uno de estos productos es el grano de chocho el valor del grano trillado y sin ningún proceso de clasificación tiene un valor por debajo de valor establecido comúnmente por los comerciantes pero ya el costal de grano seleccionado tiene un valor elevado en su comercialización.

El agricultor en la comunidad de el Abra no ha implementado dispositivos o equipos tecnológicos en procesos de producción agrícola, todos los procesos se lo han realizado de forma manual, el factor fundamental es la falta de economía, falta de conocimiento.

El objetivo principal del análisis es determinar los requerimientos del sistema de acuerdo con el software y hardware empleados en el proceso de diseño del sistema, esto implica incluir una placa embebida con requisitos establecidos y una cámara digital que permita la adquisición de datos para procesos de entrenamiento del sistema.

3.3.Diseño del sistema: Primera Iteración

En este apartado se muestra el proceso del diseño de la estructura física del sistema embebido, el lugar donde estará ubicado el equipo y los sensores; en este caso la cámara cabe mencionar que la cámara debe tener un buen ángulo de visión de los granos de Lupinus (chocho), de igual forma la parte de software del sistema, en este caso recolección de datos, entrenamiento del sistema y funcionamiento del sistema.

3.3.1. Diseño Estructural

Para la colocación del sistema embebido en necesario la construcción de una estructura física donde se aloje la placa y los sensores, en este caso la cámara debe tener un buen ángulo de visión, por lo cual la estructura debe tener dimensiones de 30 cm x 20 cm.

Figura 12

Simulación del diseño estructural del sistema

Nota. Simulación del escenario de pruebas del sistema de clasificación del grano de chocho a través de visión artificial. Fuente: Autoría propia

3.3.2. Descripción del funcionamiento del sistema embebido

El siguiente proyecto se describe como un sistema de clasificación del de grano seco Lupinus Multabilis Sweet(chocho), para lo cual se ha establecido tres características del grano, bueno, mayo y partido. Este sistema de clasificación se realizará mediante el uso de dispositivos electrónicos de bajo costo, como también una interfaz de usuario fácil de utilizar, de igual forma aplicar métodos de aprendizaje computarizado para el manejo y procesamiento de datos. Los datos obtenidos se utilizarán en una placa SBC a través de una cámara que posibilite identificar en el grupo de granos que grano es bueno, malo y partido.

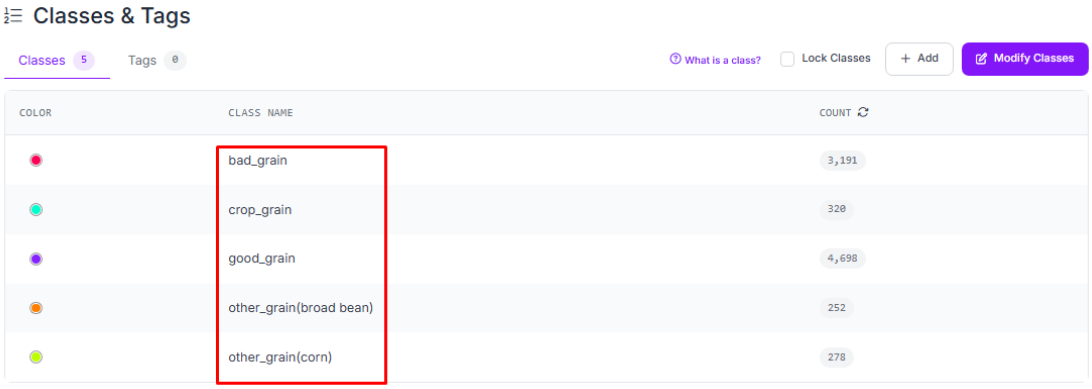
El sistema de clasificación de grano de chocho consta de dos partes como son: hardware y software, el software aquel modelo computarizado que permita el entrenamiento del sistema con un grupo de dataset previamente realizado y como parte del hardware es el sistema de alimentación y el sistema de visualización de datos resultantes.

3.3.3. Anotación y clase de grano (Chocho)

En este apartado procede a obtener las muestras del grano seco Lupinus Multabilis Sweet (chocho) de esta forma realizar el proceso de entrenamiento del sistema, para lo cual se identificado categorías o clases las cuales son: grano bueno, grano malo, grano partido, otro tipo de grano (maíz y haba). En la figura se muestra las clases que se crearon para el etiquetado del dataset para posterior el entrenamiento del sistema de clasificación del grano

Figura 13

Categoría de clasificación



Nota. Determinación de clases para el proceso de etiquetado en la plataforma de Robo Flow. Fuente: Autoría propia

3.3.3.1.Chocho Maduro (bueno)

El chocho bueno es aquel que cuenta con todas las características de un grano en buen estado como es en maduración, en coloración como también en su nivel de dureza. A continuación, en la figura se muestra un grano de chocho en buen estado.

Figura 14

Grano de chocho en buen estado

Nota. Grano de chocho en perfecto estado de madurez. Fuente: Autoría propia

Figura 15

Conjunto bueno de grano de chocho

Nota. Grano de chocho en perfecto estado de madurez. Fuente: Autoría propia

3.3.3.2.Chocho Maduro (mal estado)

El grano de chocho maduro en mal estado puede contener deformaciones, descoloraciones en su estructura física, este problema puede darse por falta de nutrientes o por exceso de agua en proceso de crecimiento o maduración. En la figura a) se muestra deformación del grano y b) descoloración del grano.

Figura 16

Grano de chocho en mal estado

Nota. Grano de chocho en mala calidad, desperfecciones físicas y decoloración en el grano. Fuente: Autoría propia

3.3.4. Chocho Maduro (grano partido)

En el proceso de trilla es posible que los granos secos tiendan a partirse por la fuerza de la maquina trilladora, por lo que al final sale granos partidos, por lo tanto, se ha considerado también crear una clase con granos partidos en el sistema embebido, en la figura 17 se muestra un grano partido.

Figura 17

Grano de chocho Partido

Nota. Grano de chocho con características de deformación o partiduras físicamente. Fuente: Autoría propia

3.3.5. Sistema de entrenamiento de datos

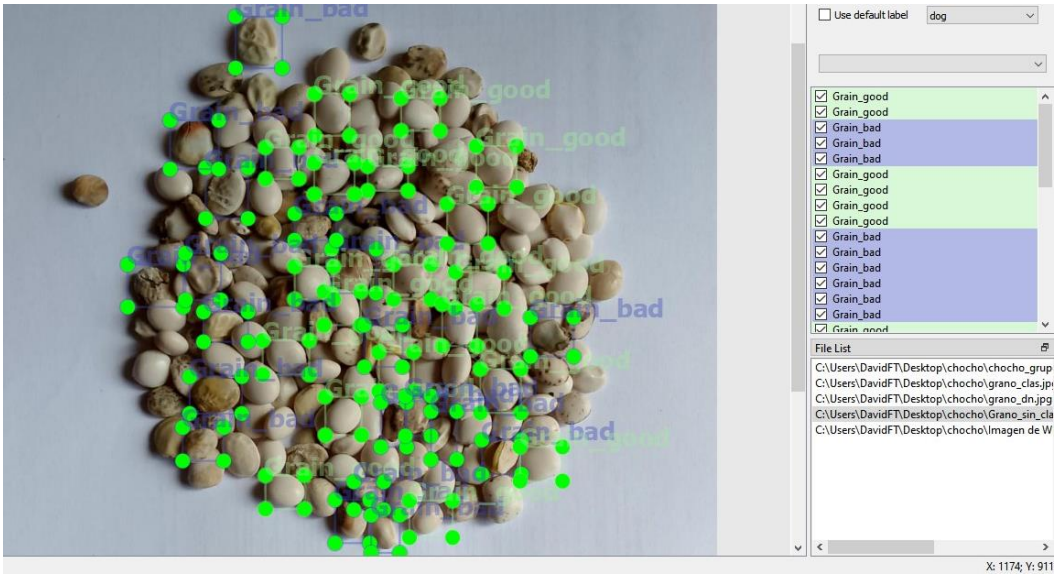
Para el entrenamiento del sistema embebido es fundamental realizar el proceso de etiquetar antes de pasar al entrenamiento de la red neuronal por lo cual sea definido en tres tipos de características como son grano bueno, grano malo y grano partido

3.3.5.1.Proceso de etiquetación de granos

Las muestras del grano de chocho pasan por un proceso de etiquetado para lo cual se ha utilizado la herramienta de código abierto LabelImg, esta herramienta cuenta con un subsistema que permite etiquetar las muestras del grano de chocho en los tres tipos de casos en la figura se muestra el proceso de etiquetado del grano en: bueno, malo y partido

Figura 18

Etiquetado del grano de chocho



Nota. Etiquetado del grano de chocho, asignación de clase como grano bueno y malo.

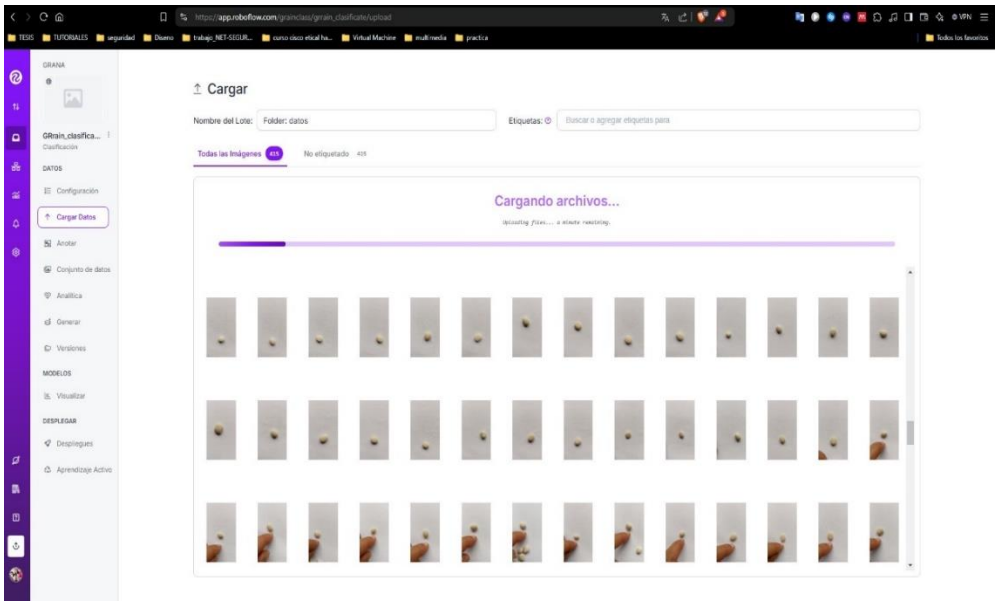
Fuente: Autoría propia

3.3.5.2.Procesamiento de imágenes

Para el procesamiento de imágenes se emplea una plataforma de código abierto Robo Flow en este se carga todas las imágenes que se tiene como dataset, se crea una cuenta para poder crear una ventana de trabajo, las imágenes cargadas son de los tres tipos de grano bueno, malo y partido, en la figura se muestra la ventana de la plataforma de Robo Flow cargando las muestras de las imágenes

Figura 19

Utilización de la plataforma Robo Flow



Nota. Carga de imágenes a la plataforma de Robo Flow para su proceso de etiquetado. Fuente: Autoría propia

3.4. Etiquetado de las muestras

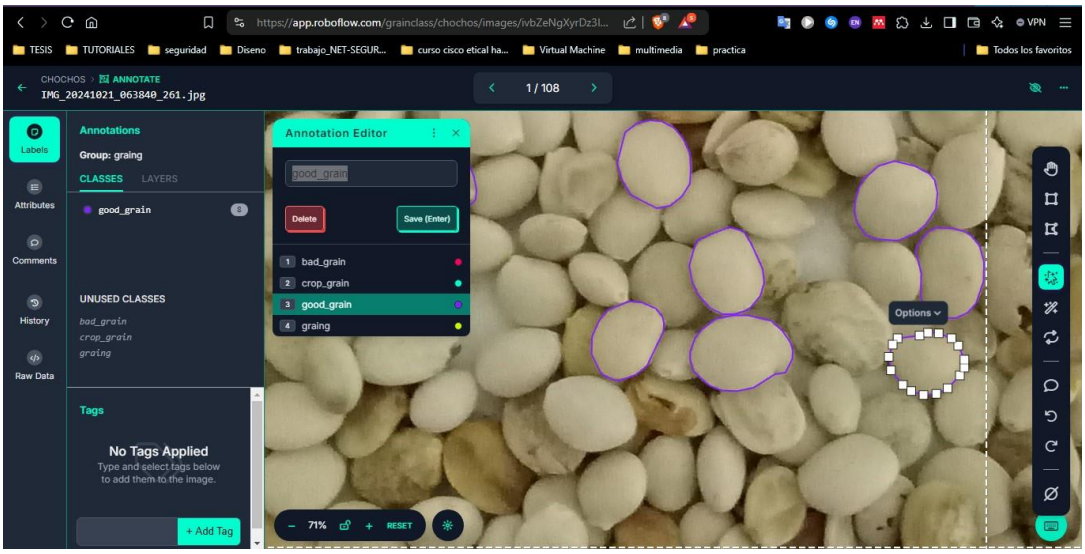
Ya con las muestras cargadas en la plataforma se procede a realizar el etiquetado de las muestras de los granos por lo que se ha establecido varias clases: grano bueno, malo, partido otro tipo de grano (maíz y haba) para lo cual se utiliza la herramienta de etiquetado proporcionada por la misma plataforma Robo Flow.

3.4.1. Etiquetado Grano Bueno

En la figura se muestra el etiquetado de los granos buenos, en la primera muestra se procede a etiquetar granos buenos y de forma individual para luego pasar el etiquetado en un montículo de granos. La herramienta de etiquetado de Robo Flow permite seleccionar de forma individual o segmentar figura por figura.

Figura 20

Etiquetado del grano de chocho en buena calidad



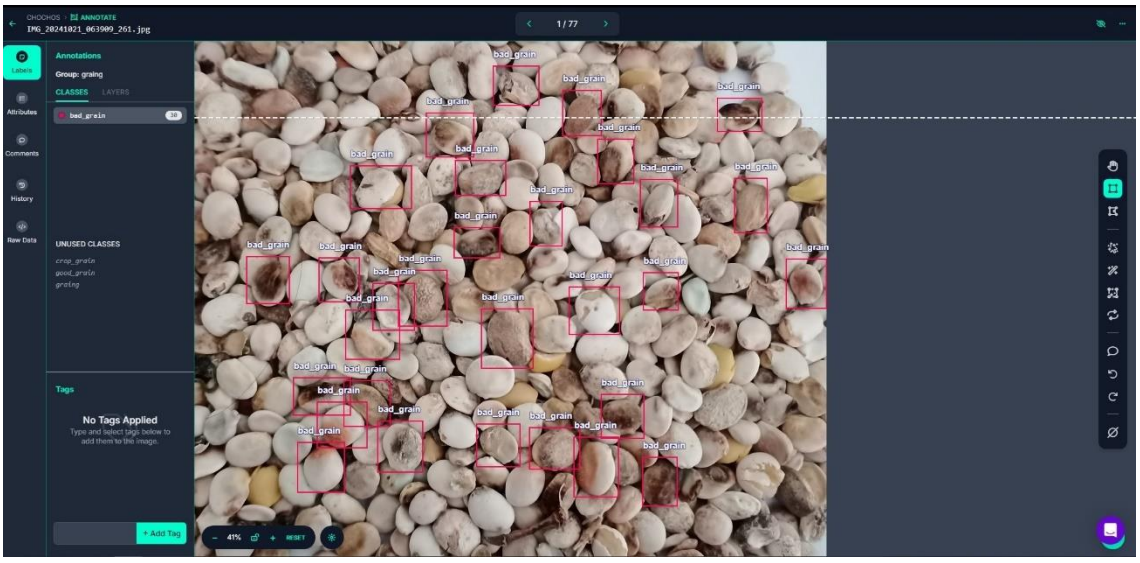
Nota. Etiquetado del grano bueno en la plataforma de Robo Flow. Fuente: Autoría propia

3.4.2. Etiquetado grano(malo)

Como siguiente clase se tiene el grano malo, de igual forma en la plataforma se cargan todas las imágenes correspondientes al grano malo y se procede a su etiquetado, también se lo realizó grano por grano y también con un montículo de granos en mal estado, una característica a tener en cuenta es que a diferencia del grano bueno en la plataforma se puede segmentar grano por grano en cambio en las muestras del grano malo no es posible ya que el grano presenta muchas imperfecciones fiscas por lo que para su etiquetado es recomendable realizarlo utilizando un cajón de selección, en la figura se muestra como es el proceso de etiquetado del grano malo en la plataforma de Robo Flow.

Figura 21

Asignación de etiqueta del grano en mal estado con la clase grano malo



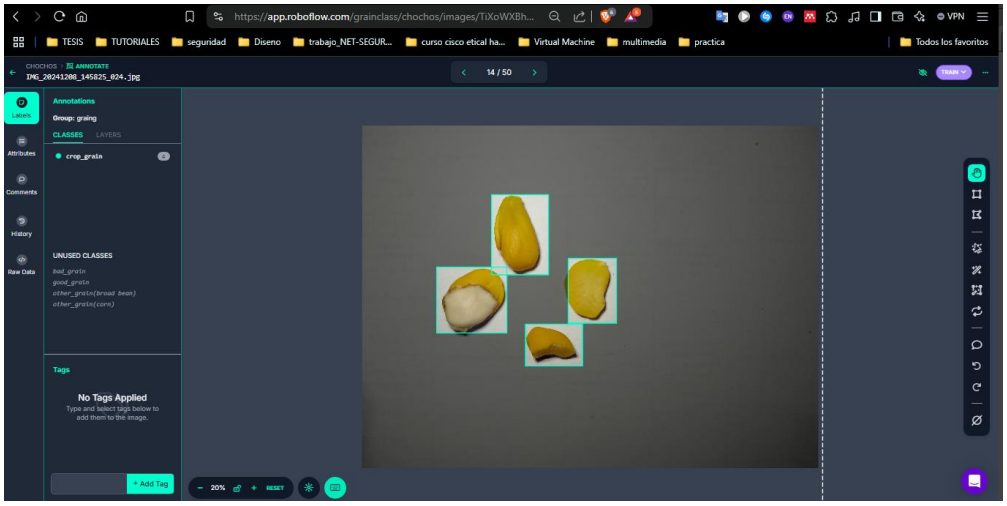
Nota. El grano de chocho malo su etiqueta se lo realiza asignado un rectángulo de selección en la plataforma de Robo Flow. Fuente: Autoría propia

3.4.3. Etiquetado grano Partido

De igual forma para la clase del grano partido se procede etiquetar en la plataforma de Robo Flow, en la figura se muestra el proceso de etiquetado del grano partido.

Figura *22*

Grano partido etiquetado en Robo Flow



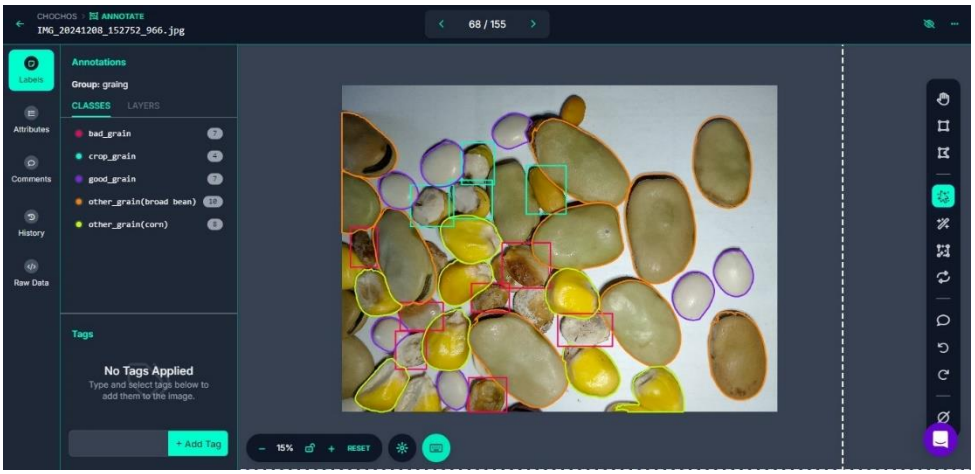
Nota. Grano partido, se identifica por falta de corteza de color blanquecina. Fuente: Autoría propia

3.4.4. Etiquetado otro tipo de grano

Para mejorar el sistema de clasificación de grano, se procedió a incluir otro tipo de granos en este caso se añadió dos clases que son otro tipo de grano: maíz y haba, mediante el uso de la herramienta de etiquetado en Robo Flow se procede a su etiquetado correspondiente, en la figura se muestra el etiquetado de los granos diferentes.

Figura 23

Proceso de etiquetado de los diferentes tipos de grano



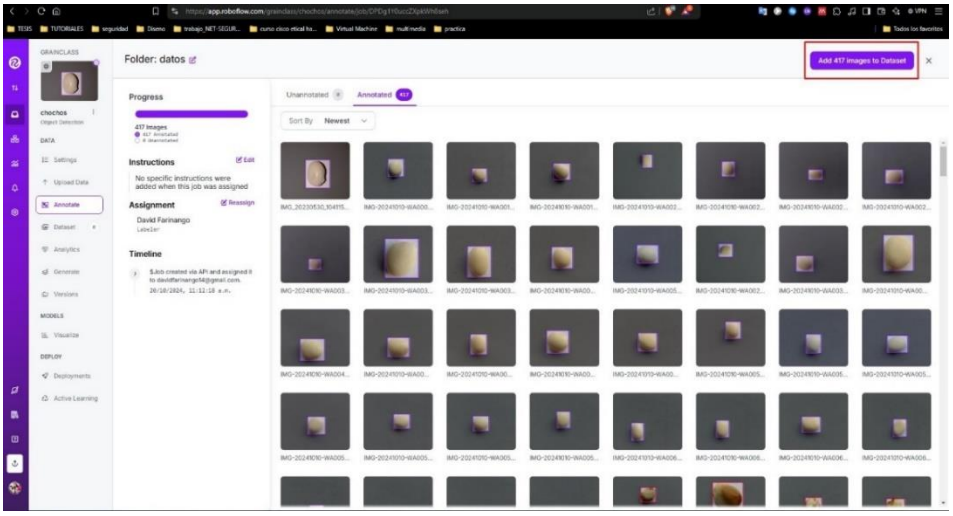
Nota. Proceso de otros tipos de grano para el sistema de clasificación del grano de Chocho. Fuente: Autoría propia

3.4.5. Crear Dataset

Luego de realizar el etiquetado correspondiente se procede a crear el dataset de las muestras que se agregaron a la plataforma de Robo Flow, en este caso como inicio se agregaron 680 muestras de grano, en la figura se muestra el grano de chocho etiquetado listo para crear el dataset.

Figura 24

Asignación de datos para el dataset

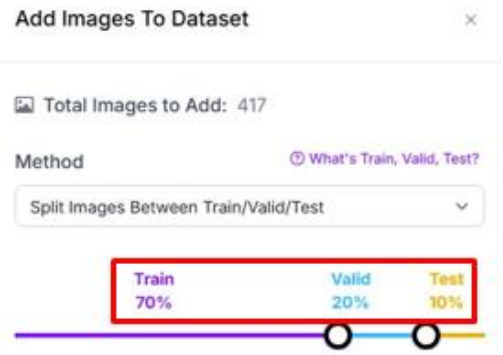


Nota. Proceso para generar el dataset del grano de chocho ya asignado las diferentes clases. Fuente: Autoría propia

Ya creada el dataset se procede a realizar el entrenamiento, para lo cual en la plataforma de Robo Flow se añade porcentajes de entrenamiento, validación y prueba, la figura muestra el porcentaje que se utilizara para el entrenamiento.

Figura 25

Datos para el entrenamiento



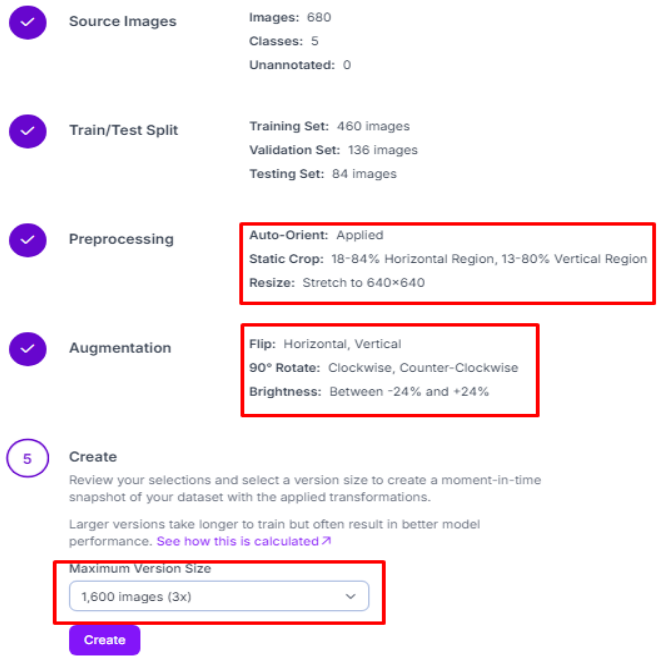
Nota. Proceso de entrenamiento con el dataset asignado, parámetros establecidos por el sistema. Fuente: Autoría propia

En la plataforma RoboFlow se especifica la creación del dataset, menciona la integración de las muestras para inicio del entrenamiento dando como resultado una versión de datos entrenado, añade información de que tipos de clases se ha creado, la figura indica las características del entrenamiento.

Se cambia los parámetros de entrenamiento, modificando la rotación de la imagen, enfoque de imagen, saturación de imagen. En la figura se muestra los parámetros ya configurados y muestra el total de dataset que se creara en esta versión.

Figura *26*

Cambio de parámetros para generar un mayor número de dataset.



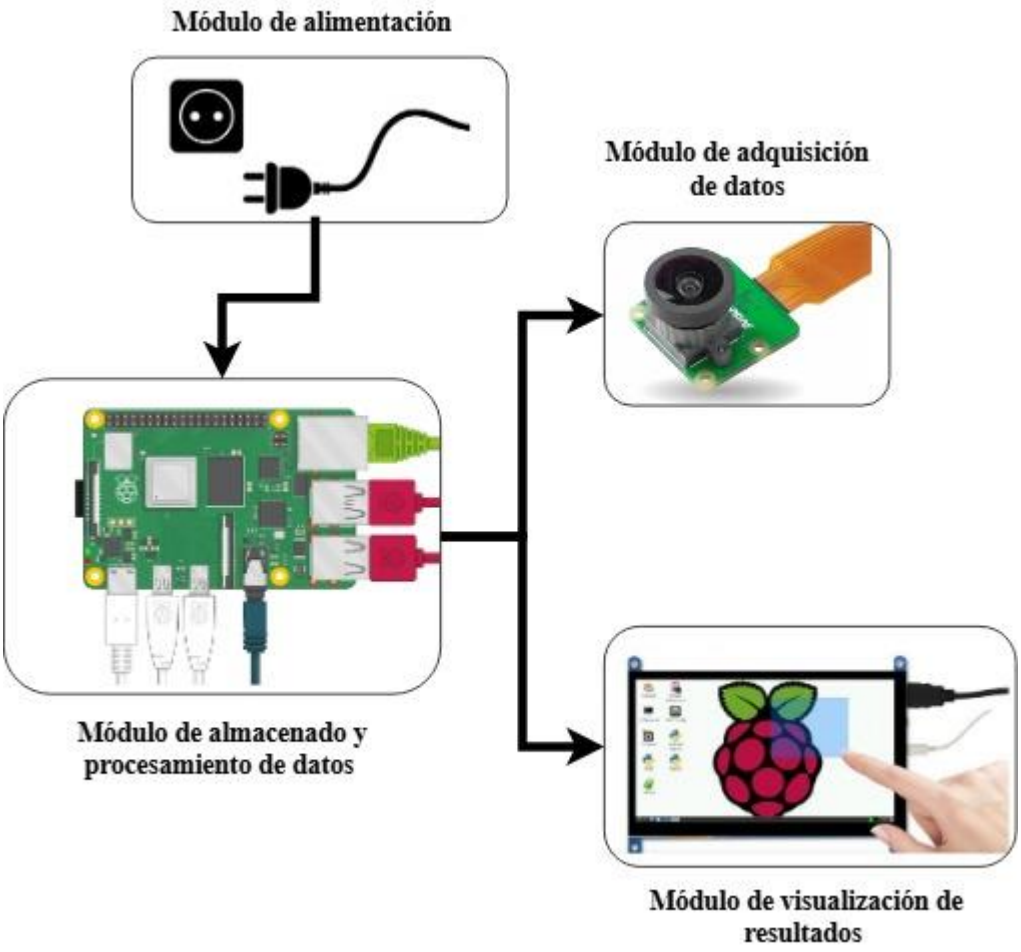
Nota. Cambio de parámetros de etiquetado en la plataforma de Robo Flow. Fuente: Autoría propia

3.5.Diagramas del sistema

De forma general es necesario detallar el funcionamiento del proyecto a través de graficas que muestren los procesos que realiza en cada etapa por lo cual en la figura27 se identifica de manera general a través de diagramas de bloques la conexión del sistema embebido.

Figura 27

Diagrama de funcionamiento del sistema



Nota. Diagrama del sistema embebido de clasificación del grano de chocho a través de visión artificial. Fuente: Autoría propia

En el módulo de alimentación se encuentra toda la conexión hacia la red eléctrica este puede ser un cargador el cual encienda en este caso a la placa SBC o también un banco de energía configurado para suministrar la energía necesario para encender la placa.

El módulo de adquisición de datos corresponde a un sensor(cámara) que permite capturar imágenes del grano de chocho conforme a las clases ya predefinidos porque es fundamental asignar una área determinado, este ubicado de acuerdo con el diseño del sistema.

Al módulo de almacenamiento y procesamiento de datos se definió al equipo que permita la ejecución de la arquitectura YOLO y la ejecución del modelo ya entrenado para la clasificación del grano de chocho.

Finalmente, como módulo de visualización es el encargado de mostrar el resultado obtenido luego de que la información haya pasado por el módulo de procesamiento. Como resultado se tendrá grano bueno, malo, partido y otro tipo de grano.

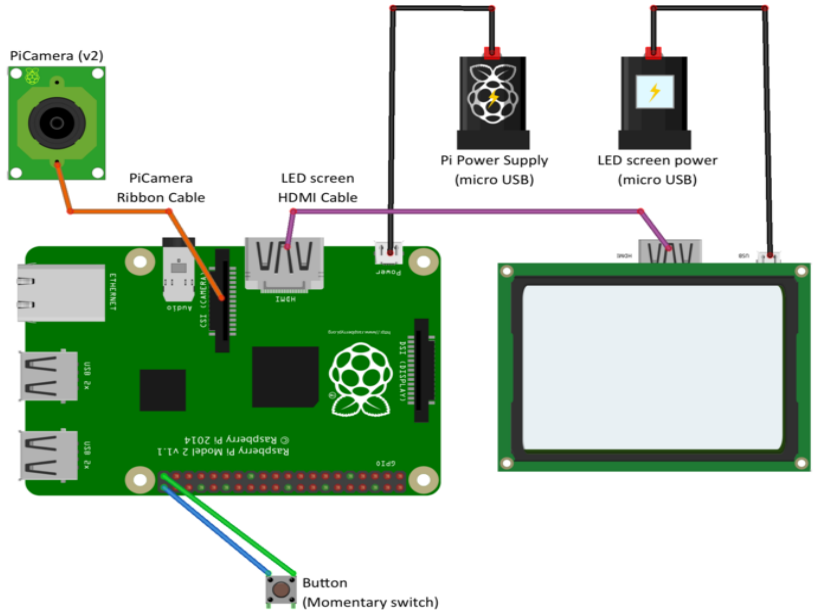
En la siguiente figura se muestra el funcionamiento del sistema embebido de manera lineal, detallando cada etapa desde la adquisición de imágenes, etiquetado, procesamiento, entrenamiento y finalmente visualización de datos para el sistema de clasificación del grano de chocho.

3.6. Conexión de Componentes

En este apartado se detalla la simulación correspondiente a la conexión de los componentes con la placa SBC Raspberry PI5, la cámara PI es la encargada de obtener toda la cantidad de información en este caso fotografías de los granos de chocho, ya obtenido la fotografía se pasa a la placa SBC PI5 para su procesamiento, el modelo entrenado Yolo valida que tipo de grano se muestra y lo pasa a visualizar en la pantalla. En la figura 28 se muestra la conexión de componentes del sistema.

Figura 28

Conexión de componentes electrónicos a la placa SBC.



Nota. Diagrama de conexión de componentes del sistema. Fuente: Autoría propia

3.7. Conexión física de dispositivos

Se realizó la conexión física de los componentes del sistema embebido para lo cual se utilizó una Raspberry PI5, una Arducam IMX477, una pantalla LCD de 9 pulgadas, una fuente de 5V a 3A, un ventilador que permita la refrigeración del equipo central. En la figura 30 de evidencia la conexión física de los dispositivos.

Figura 29

Conexión física del sistema embebido.

Nota. Diagrama físico de conexión de componentes del sistema embebido. Fuente: Autoría propia

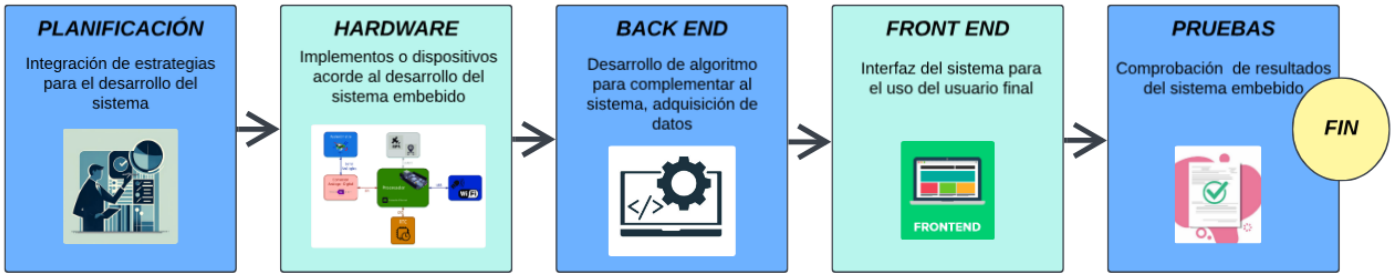
Figura 30

Diagrama lineal de funcionamiento

Nota. Diagrama lineal del sistema. Fuente: Autoría propia

Figura 31

Diagramas de etapas operativas del sistema embebido



Nota. Diagrama de etapa operativa del sistema. Fuente: Autoría propia

3.7.1. Diagramas de flujo

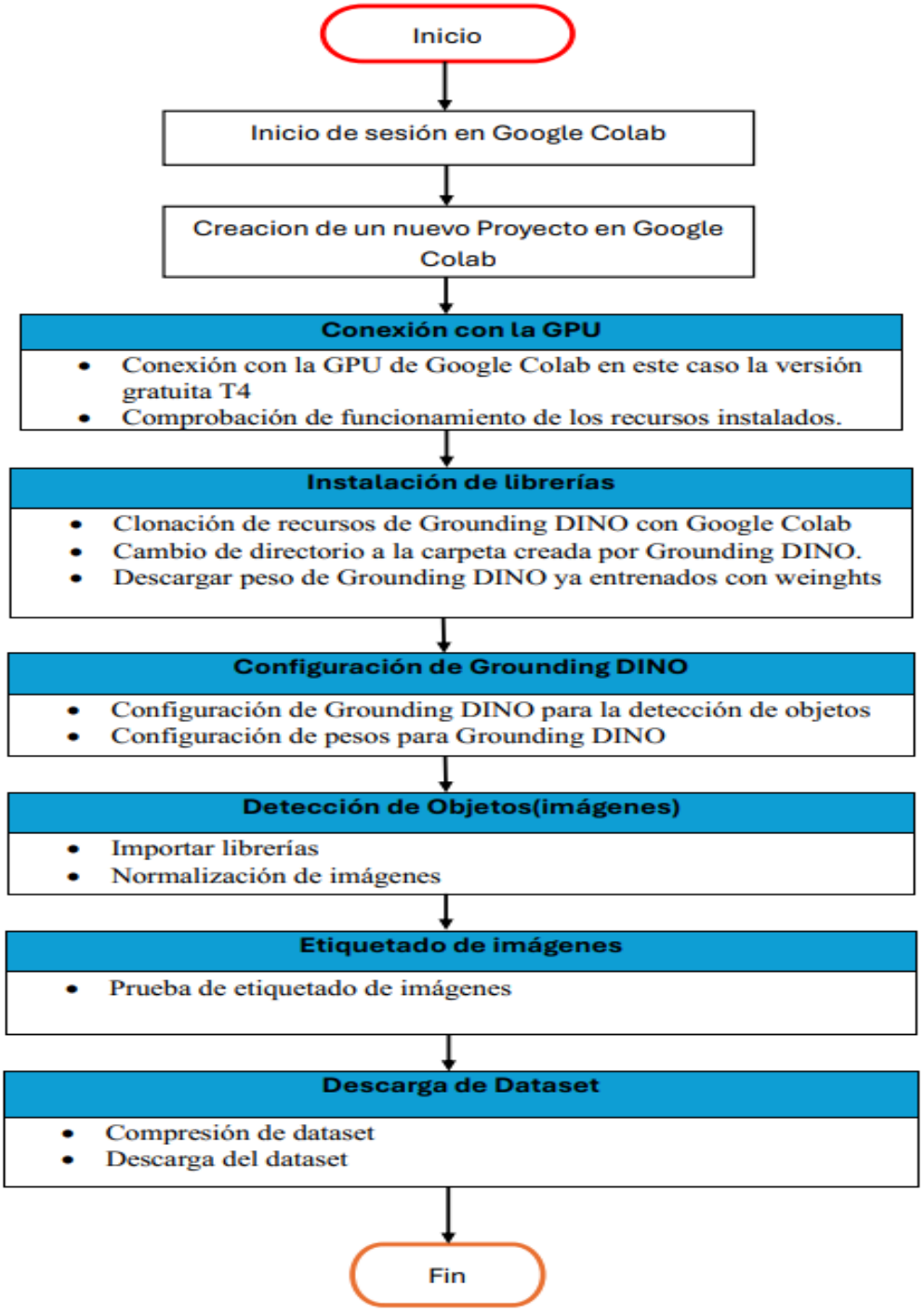
En este apartado se procede a describir los procesos que contiene cada etapa, en cada etapa se ejecuta tareas específicas desde la adquisición, validación y visualización de datos, por lo tanto, es fundamental detallar cada etapa en diagramas de flujo, mediante este detallar el funcionamiento correcto del sistema.

3.7.1.1. Flujograma del proceso de etiquetado del dataset (Gounding DINO)

El proceso de entrenamiento del sistema consiste en recopilar una gran cantidad de data para formar un dataset adecuado para el tipo de clasificación que se requiera realizar en este caso la data correspondiente al tipo de imágenes de los granos de chochos, para el proyecto la data se clasifica en grano bueno, grano malo, y grano partido.

Figura *32*

Flujograma de funcionamiento del sistema de etiquetado



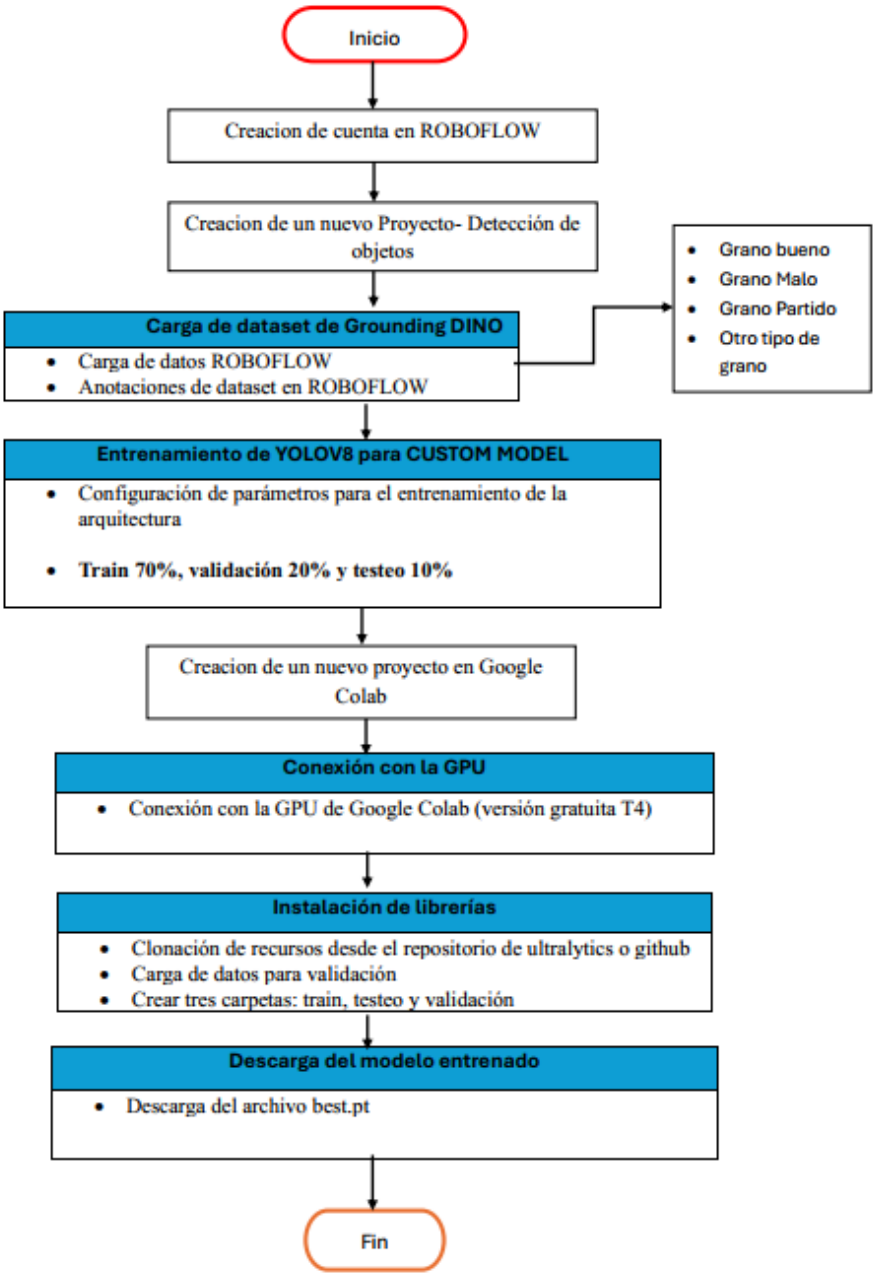
Nota. Flujograma de etiquetado. Fuente: Autoría propia

3.7.1.2. Flujograma de procesos de entrenamiento del modelo (YOLOV8)

Ya establecido el dataset de los componentes utilizados se procede a entrenar el modelo utilizando la arquitectura de yolov8 de la familia (YOLO) esta arquitectura es la última en la interacción por lo tanto el dataset utilizado para la detección de granos aún no existe en los repositorios de YOLO, en la siguiente figura se muestra de manera estructurada como es el proceso de entrenamiento de la arquitectura yolov8 para la detección de grano de chocho postcosecha.

Figura 33

Flujograma de la arquitectura YOLOv8



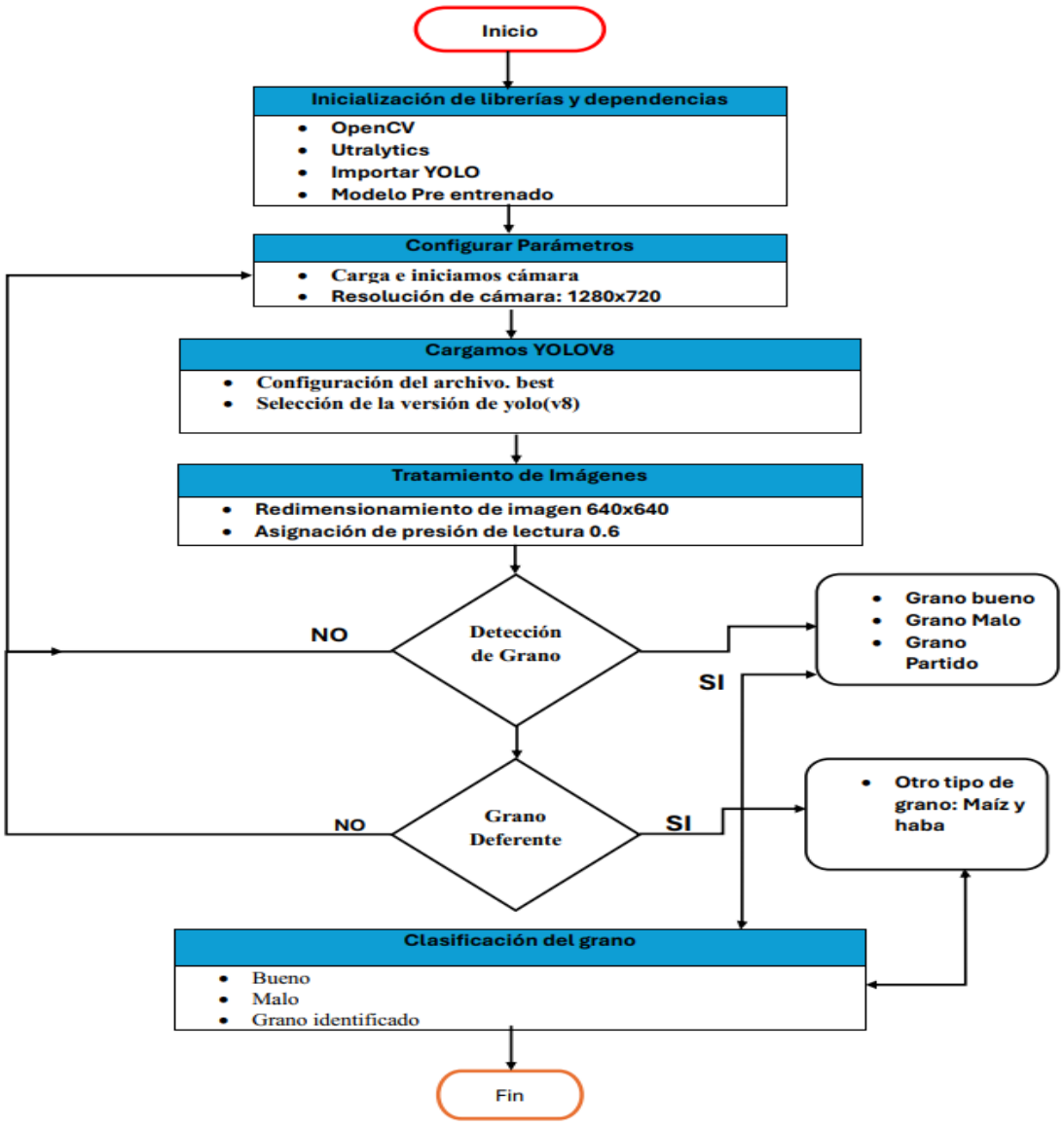
Nota. Flujograma de funcionamiento de YOLOv8. Fuente: Autoría propia

3.7.1.3. Flujograma de inferencia del sistema embebido.

A continuación, se muestra el flujograma de la integración del sistema embebido, en la figura se detalla cada uno de los procesos realizados en la integración del sistema.

Figura 34

Flujograma de inferencia del sistema



Nota. Flujograma de inferencia. Fuente: Autoría propia

3.8. Entrenamiento Red Neuronal

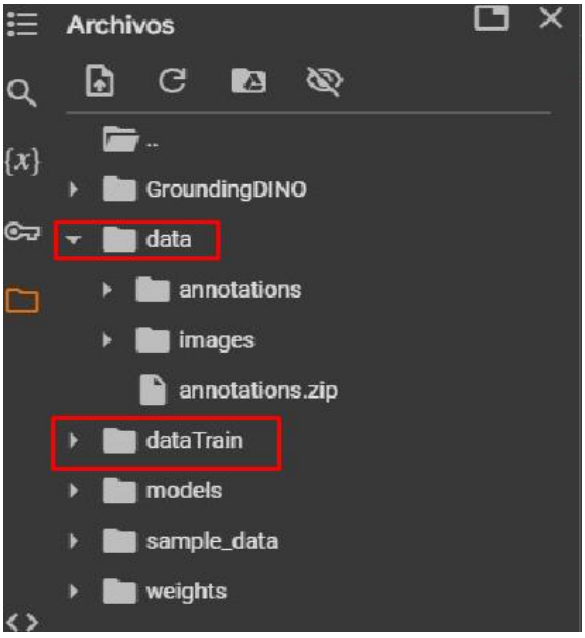
Para el entrenamiento de la arquitectura Yolo en su versión V8, se utilizó plataformas gratuitas en este caso herramienta de desarrollo de Google como es Google Colab, desde esta herramienta se procedió a generar el entrenamiento del modelo para posterior realizar la inferencia de todo el sistema.

3.8.1. Script Modelo YOLO(v8)

En un nuevo proyecto de Google Colab se procede a crear varios directorios con el fin de almacenar los dataset de las clases ya definidos como son grano bueno, grano malo y grano partido, en cada uno de los directorios esta un subdirectorio identificado como entrenamiento, testeo y validación, en la figura se muestra el directorio creado en el proyecto de Google Colab.

Figura 35

Creación de directorios para prueba de funcionamiento.

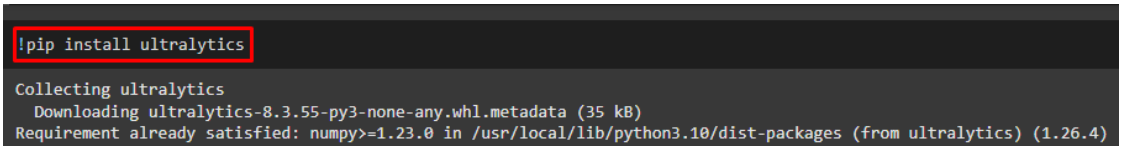


Nota. Directorio de etiquetado y prueba. Fuente: Autoría propia

Luego de crear los directorios correspondientes se procede a instalar la dependencia de Yolo en este caso la librería de Ultralytics, en la figura se muestra el comando utilizado para la instalación de las librerías que corresponde a Ultralytics.

Figura 36

Instalación de librerías

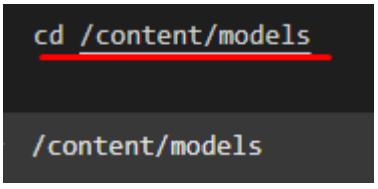


Nota. Descarga de librerías. Fuente: Autoría propia

Con la librería instalado correctamente se procede a ingresar a los directorios que ya se crearon con anterioridad para luego proceder a instalar el modelo Yolo en su octava versión siendo la más recientes en sus actualizaciones. En el directorio ya creado se crea un subdirectorio con el nombre de models

Figura 37

Cambio de directorio para prueba

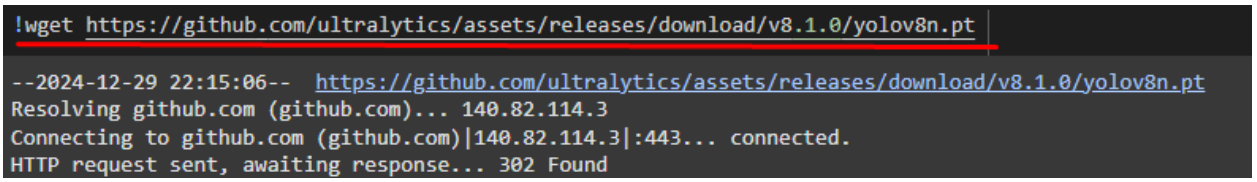


Nota. Directorio principal de archivos. Fuente: Autoría propia

En el directorio de models se procede a clonar desde los repositorios de GitHub la versión más reciente de YOLO en este caso la versión 8, en esta interacción mejora la detección de objetos

Figura 38

Descarga de Yolov8 desde el repositorio

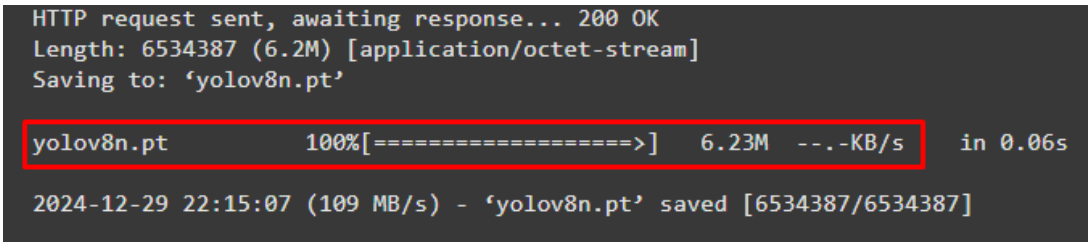


Nota. Descarga de la versión de Yolov8. Fuente: Autoría propia

Ya confirmado la respuesta desde el repositorio se procede a descargar la versión de YOLO, en este caso la última versión la cual la octava, en la figura se muestra el tamaño del archivo que se descargó desde el repositorio.

Figura 39

Versión de Yolov8 descargado



Nota. YoloV8 descargado correctamente. Fuente: Autoría propia

Capitulo IV

EJECUCIÓN Y PRUEBAS DE FUNCIONAMIETO

En este capítulo se evidencia el proceso que se llevó a cabo en la ejecución del sistema embebido para la clasificación del grano de chocho, mostrando los resultados obtenidos luego de las pruebas que corresponde para su adecuado funcionamiento.

4.1. Diagrama de Funcionamiento

Para comprender de mejor manera como es el funcionamiento del sistema se ha realizado una topología como se aprecia en la figura. El sistema embebido inicia con la captura de imágenes del grano de chocho en una bandeja para luego pasar el siguiente proceso que es la ejecución del modelo entrenado, este modelo se estará ejecutando desde una placa embebido Rasberry PI5, como siguiente paso es la ejecución de la arquitectura de la red neuronal la cual es YOLO en su octava versión para lo cual se utilizó la plataforma de Google Colab para el entrenamiento de la arquitectura.

4.2 Pruebas de funcionamiento

Para la determinar el funcionamiento del sistema se procedió a realizar las respectivas pruebas, en este caso como la plataforma de ROBOFLOW está compuesta con varias herramientas, una de estas es la verificación del modelo a través de una imagen o directamente se descarga una aplicación para realizar el testeo desde un dispositivo móvil.

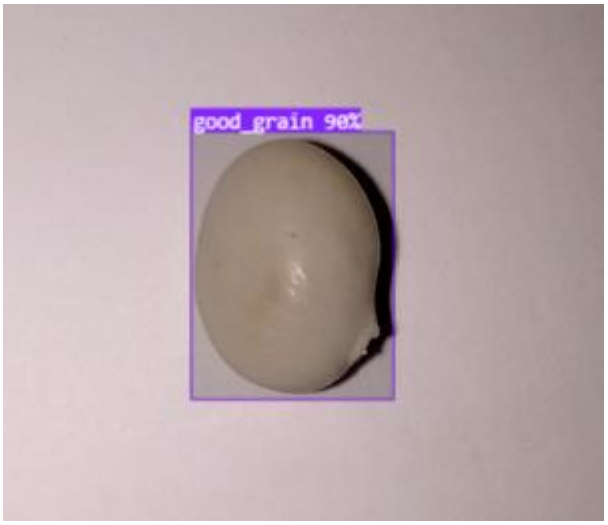
4.1.1. Grano Bueno

Para el grano de chocho en buen estado y de buena calidad la prueba se lo realizo de manera individual es decir grano por grano y también formando un conjunto o montículo de grano bueno

en la siguiente figura 38 se muestra el funcionamiento del modelo yolov8 entrenado para la identificación de un grano

Figura 40

Identificación del grano bueno

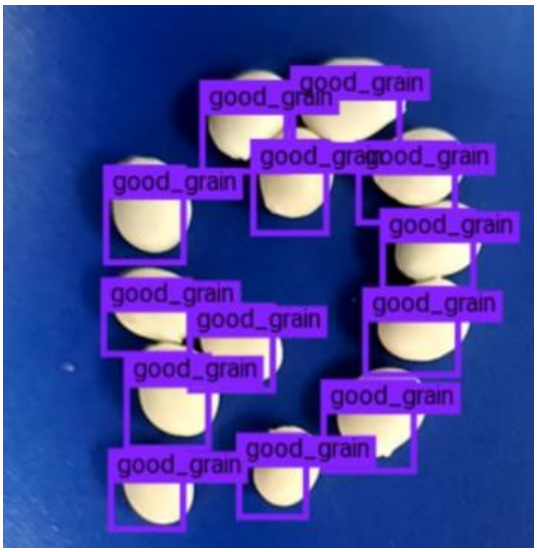


Nota. Grano de chocho en buen estado. Fuente: Autoría propia

En la siguiente figura 40 se realiza la prueba para un conjunto de granos y el sistema reconoce y muestra las etiquetas correspondientes a grano bueno.

Figura 41

Grano de chocho identificado como grano bueno

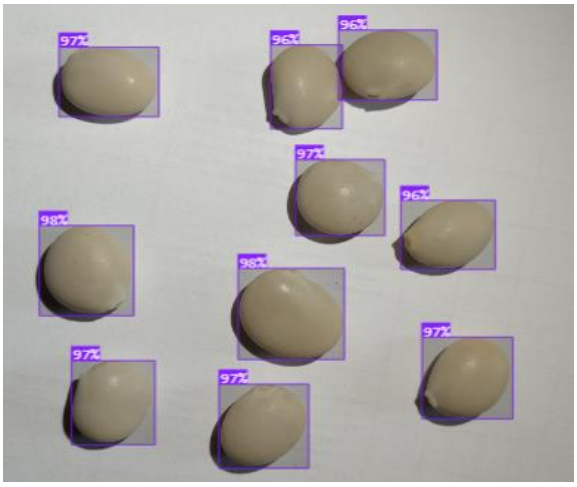


Nota. Conjunto de grano de chocho en buen estado. Fuente: Autoría propia

De acuerdo con valores de precisión para la identificación del grano se tiene un valor en porcentaje por encima del 95%, este valor depende a los valores de precisión del modelo en el entrenamiento para este caso el valor de precisión se quedó establecido en 73%

Figura 42

Precisión de reconocimiento del gano



Nota. Porcentaje de precisión de grano de chocho en buen estado. Fuente: Autoría propia

4.1.2. Grano bueno con otro tipo de clase

Con las diferentes clases ya entrenados y funcionado el sistema es capaz de identificar cual es el grano de la clase grano bueno, malo y partido, en la figura 42 se muestra la identificación de los granos.

Figura 43

Grano de chocho en diferentes clases

Nota. Grano de chocho identificado con diferentes etiquetas. Fuente: Autoría propia

4.1.3. Grano Malo

Las pruebas realizadas para determinar el grano malo de entreno al sistema con diferentes tipos ya que, a diferencia del grano bueno, este grano presenta varias deformaciones, descoloraciones por lo que es dificultoso para el sistema diferenciar del grano bueno o malo, el entrenamiento del sistema para determinar el grano malo se lo realizo con un número mayor de dataset que para el grano bueno. En la siguiente figura se muestra el sistema identificando el grano malo.

Figura 44

Identificación de grano malo de chocho

Nota. Grano de chocho identificado como grano malo. Fuente: Autoría propia

De igual forma al grano malo se lo identifico correctamente por el sistema la correcta identificación es medida a un valor de 95% de reconocimiento del grano malo, en la figura se muestra cómo se identifica al grano.

Figura 45

Grano malo detallado según su precisión

Nota. Porcentaje de precisión de identificación. Fuente: Autoría propia

4.1.4. Grano Partido

El sistema también fue entrenado con la clase de grano partido mediante la cual se identificó granos con desfiguramiento en la corteza por lo tanto se procedió a etiquetar como grano partido, en la figura se muestra la identificación correcta por parte del sistema.

Figura 46

Detalles del grano partido

Nota. Grano de chocho identificado como grano partido. Fuente: Autoría propia

4.1.5. Prueba con diferentes clases de grano

Se procedió a juntar todos los granos como buenos y malos para determinar el funcionamiento del sistema, la identificación del grano se lo realizo con un Bouding Box que marca que tipo de grano se encuentra, en la figura se muestra en funcionamiento del sistema.

Figura 47

Determinación de grano bueno y malo ya con las clases de grano asignado

Nota. Sistema identificando grano bueno o malo. Fuente: Autoría propia

4.1.6. Otro tipo de grano

El sistema también fue entrenado en detectar otro tipo de grano, por lo tanto, se procedió a etiquetar el grano diferente, para esto también de utilizo la aplicación de Robo Flow, en este caso el grano diferente también se agregó a otra clase llamada otro tipo de grano (maíz), en la figura se muestra la identificación de otro tipo de grano.

Figura 48

Identificación de otro tipo de grano

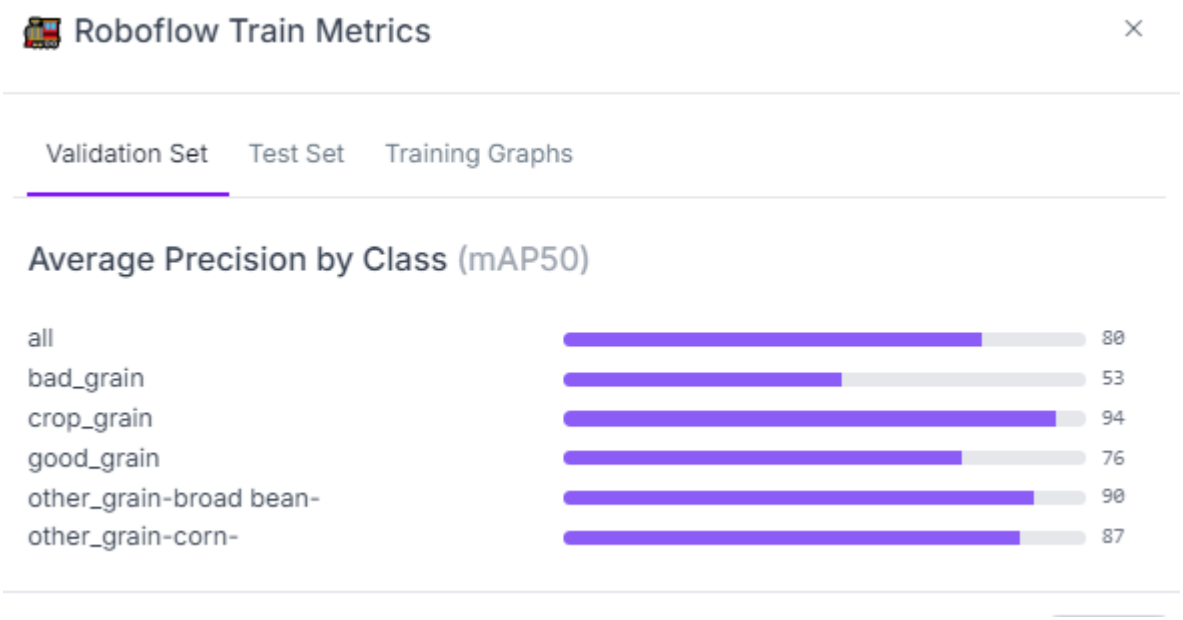
Nota. Sistema identificando de otro tipo de grano. Fuente: Autoría propia

4.3. Resultados obtenidos del sistema

De acuerdo con la plataforma de Robo Flow se detalla todos los datos obtenidos con el procesamiento de datos con las clases que se definió con anterioridad para etiquetar cada uno de granos ingresados, en Robo Flow se realizó tres interacciones para mejorar la precisión en la detección de los granos de chocho mediante el uso de visión artificial por lo que en la primera interacción se tiene una precisión de 50%, en la segunda interacción bajo el porcentaje de precisión a 48% y finalmente en la tercera interacción el valor de la precisión se obtuvo del 73% y se dejó en este margen para la detección del grano de chocho. En la figura se muestra los resultados obtenidos en la validación de etiquetas de todas las clases obtenidos en entrenamiento.

Figura 49

Detalles de validación de datos



Nota. Resultados de validación de datos. Fuente: Autoría propia

4.4. Análisis de costo Beneficio

Para el desarrollo del sistema embebido de clasificación de grano de chocho se ha realizado un análisis detallado se todos los componentes que fueron parte del proceso de construcción dentro de este grupo de análisis se determina el costo de hardware, software, infraestructura e ingeniería y finalmente describir el beneficio obtenido del funcionamiento al finalizar el sistema embebido de clasificación del grano de chocho.

4.4.1. Costos de Hardware

El costo por cada equipo utilizado en el desarrollo se detalla en la siguiente tabla, se describe el valor unitario de cada componente como también el valor general de todos los componentes de hardware utilizados en el sistema.

Tabla 15

Costo de Hardware

Costos de Hardware			
Cantidad	Hardware	Costo Unitario	Costo Total
1	Cámara Raspberry Pi HQ de alta calidad	$68,40	$68,40
1	Raspberry pi5	$145	$145
1	Micro SD	$20	$20
1	Monitor	$130	$130
	Total		$363.40

Nota. Costos de componentes de hardware. Fuente: Autoría propia

4.4.2. Costos de software

Se detalla los valores en costo referentes a software empleado en el desarrollo del sistema, en la tabla 16 se muestra detallado de forma unitario como también el costo total el costo del software, cabe mencionar que en su mayoría se ha empleado softwares de código abierto para minimizar los costos.

Tabla 16

Costos de Software

Costos de Software			
Cantidad	Software	Costo Unitario	Costo Total
1	Google Colab	$0	$0
1	Python	$0	$0
1	Google Drive	$0	$0
1	Pycharm	$0	$0
	Total		$0

Nota. Costo de software. Fuente: Autoría propia

4.4.3. Costo de infraestructura

Los costos incluidos como infraestructura son los costos por los materiales extras que se emplean en la construcción del sistema, para el proyecto se empleó piezas de madera como soporte de cámara y de la placa, en la tabla se muestra todos los costos de infraestructura empleados en la construcción del sistema.

Tabla 17

Costo de infraestructura

Costos de Infraestructura			
Cantidad	Elemento	Costo Unitario	Costo Total
4	Madera	$2	$8
6	Leds	$1.25	$7.50
1	Bandeja	$3	$3
	Total		$18.50

Nota. Costo de infraestructura. Fuente: Autoría propia

4.4.4. Costo de ingeniería

En los costos de ingeniería es fundamental detallar cada una de las actividades realizados en el diseño del hardware, recopilación de dataset para las diferentes clases definidos, entrenamiento del sistema y codificación para la inferencia del modelo Yolo V8, en la tabla se muestra los costos asociados a costos de ingeniería.

Tabla 18

Costos de Ingeniería

Costos de Ingeniería		
Cantidad	Costo Unitario	Costo Total
Estudio de área y documentación	$30	$30

Deseño de Hardware, programación de código de entrenamiento.	$200	$200
Implementación de infraestructura.	$100	$100
	Total	$330

Nota. Costo de ingeniería. Fuente: Autoría propia

4.4.5. Costo general del sistema

En el desarrollo del sistema embebido ha implicado varios costos desde hardware, software, infraestructura y también costos de ingeniería. Es por lo tanto detallar de manera general los costos totales. En la tabla 19 se menciona todos los costos generados por parte del proyecto.

Tabla 19

Costos generales del sistema

Costos General	
Descripción	Costo Unitario
Costos de Hardware	$363.40
Costos de Software	$0
Costo de Infraestructura	$18.50
Costos de Ingeniería	$330
TOTAL	$711.90

Nota. Costos generales en el desarrollo del proyecto. Fuente: Autoría propia

4.4.6. Beneficios

En este apartado se detallan los beneficios del sistema embebido de clasificación del grano de chocho a través de visión artificial y utilización de técnicas de aprendizaje profundo, a continuación, se describen los beneficios obtenidos:

- Implementación de tecnología en el sector agrícola de esta manera permitir mejoras al agricultor en las actividades que realizan en el proceso de selección para el grano de Lupinus Mutabilis Sweet (chocho) postcosecha.
- Determinación de la calidad del grano Lupinus Mutabilis Sweet (chocho) al momento de identificación y clasificación mediante un sistema embebido previamente entrenado.
- El sistema elimina el criterio de identificación y clasificación manual, que requiere dependencia de mucho tiempo, mano de obra costosa y propensa a errores. Este sistema provee procesar grandes volúmenes de granos en menos tiempo con la implementación de un sistema industrial y de esta forma aumentando la eficiencia en la cadena productiva.
- El diseño modular del sistema de clasificación de grano permite adaptarlo para clasificar otros tipos de granos (maíz, haba) con mínimas modificaciones. Además, puede escalarse para operar en entornos industriales agrícolas.
- El sistema embebido da paso al acceso a tecnologías avanzadas como la visión artificial y el aprendizaje automático, que tradicionalmente están implementados en grandes empresas. Esto ayuda a los pequeños productores a mejor su eficiencia en el proceso de comercialización en el mercado.

Conclusiones

El desarrollo de Diseño de un sistema embebido de clasificación del grano de Lupinus Mutabilis Sweet (chocho) mediante el uso de visión artificial ha demostrado ser una solución eficiente y viable para optimizar procesos agrícolas. A lo largo de este trabajo, se logró diseñar e implementar un prototipo capaz de identificar el grano de chocho con base en características como tamaño, color y defectos físicos, utilizando técnicas de procesamiento de imágenes y algoritmos de inteligencia artificial.

Entre los principales hallazgos, se destaca la precisión del sistema en la clasificación, alcanzando un porcentaje de acierto del 73%, lo que evidencia su potencial para ser implementado en entornos industriales. Además, el uso de hardware embebido permitió reducir costos y mejorar la portabilidad del sistema, haciéndolo accesible para pequeños y medianos productores.

Durante el proceso se identificaron limitaciones, como la dependencia de condiciones de iluminación controladas y la necesidad de un entrenamiento más extenso del modelo de clasificación para mejorar su robustez frente a variaciones en los granos. Estos aspectos representan oportunidades para futuras investigaciones, como la integración de sensores adicionales o la implementación de algoritmos más avanzados de aprendizaje profundo.

Recomendaciones

Es fundamental ampliar el conjunto de datos utilizados para el proceso de entrenamiento del modelo de visión artificial, incluyendo una mayor variedad de granos de Lupinus Mutabilis Sweet (chocho) con diferentes características (tamaños, colores y defectos físicos). Esto permitiría optimizar la precisión y robustez del sistema frente a pruebas en campo real.

Para obtener una calidad adecuada de las imágenes capturadas es necesario implementar un sistema de iluminación controlada o integrar sensores que ajusten automáticamente los parámetros de luminosidad para mejorar la captura de imágenes para su procesamiento en el sistema de acuerdo con las condiciones ambientales.

Para mejorar la eficiencia del sistema de clasificación de grano de Lupinus Mutabilis Sweet (chocho) mediante el uso de vision artificial, se recomienda explorar el uso de hardware embebido con mejores prestaciones, como unidades de procesamiento gráfico (GPU) integradas mayor memoria RAM, que permitan ejecutar algoritmos más complejos en tiempo real.

Es necesario realizar pruebas de funcionamiento del sistema en condiciones reales de campo, donde se puedan evaluar su desempeño frente a condiciones ambientales como la humedad polvo y variaciones en la disposición de los granos de Lupinus Mutabilis Sweet (chocho).

Trabajar juntamente con especialistas en agronomía permitiría adecuar los criterios de clasificación según estándares de calidad específicos para el grano de Lupinus Mutabilis Sweet (chocho), asegurando que el sistema cumpla con las necesidades reales del sector agrícola y comercialización del grano.

Bibliografía

Agustian, D., Pertama, P. P. G. P., Crisnapati, P. N., & Novayanti, P. D. (2021). Implementation of Machine Learning Using Google's Teachable Machine Based on Android. *3rd International Conference on Cybernetics and Intelligent Systems, ICORIS 2021*. https://doi.org/10.1109/ICORIS52787.2021.9649528

Basantes, F., Aragón, J. C., & Albuja, M. (2022). *Cultivos Andinos de Importancia agro-productiva y comercial en la zona 1 del ecuador*. http://repositorio.utn.edu.ec/handle/123456789/12828

Boschetti, A., & Massaron, L. (2018). *Python data science essentials: a practitioner's guide covering essential data science principles, tools, and techniques* (Vol. 3ro).

Brownlee Disclaimer, J. (2017). *Deep Learning for Natural Language Processing Develop Deep Learning Models for Natural Language in Python Acknowledgements Copyright Deep Learning for Natural Language Processing*.

Cabrera Ovando, D., Mercado Andia, K., Vargas Guzmán, D., & Julia Fernández, M. (2020). *Los caminos del tarwi y la integración andina: Bolivia, Perú y Ecuador*. *v1*, 4–24. https://interaprendizaje.ipdrs.org/images/Documentos2018/MEMORIACAMINOSDELTARWI_14.12.18.pdf

Chalampuente Doris, Tapia Cesar, & Sorensen Marten. (2021). *(1) (PDF) The Andean Lupine-'El Chocho' or "Tarwi" (Lupinus Mutabilis Sweet) General Observations on the Culture of Lupine (Lupinus Mutabilis Sweet)*. https://www.researchgate.net/publication/353185981_The_Andean_Lupine-%27El_Chocho%27_or_%27Tarwi%27_Lupinus_Mutabilis_Sweet_General_Observations_on_the_Culture_of_Lupine_Lupinus_Mutabilis_Sweet

Choy, G., Khalilzadeh, O., Michalski, M., Do, S., Samir, A. E., Pianykh, O. S., Geis, J. R., Pandharipande, P. V., Brink, J. A., & Dreyer, K. J. (2018). Current Applications and Future Impact of Machine Learning in Radiology. *Https://Doi.Org/10.1148/Radiol.2018171820*, *288*(2), 318–328. https://doi.org/10.1148/RADIOL.2018171820

Cox, T. (2016). *Raspberry Pi for Python programmer's cookbook: over 60 recipes that harness the power of the Raspberry Pi together with Python programming and create enthralling and captivating projects* (2do ed., Vol. 2).

De, C. ′, & Ordoba. (2019). *Métodos de aprendizaje automático aplicados a problemas cosmológicos*. https://rdu.unc.edu.ar/handle/11086/14284

Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., & Socher, R. (n.d.). *Deep learning-enabled medical computer vision*. https://doi.org/10.1038/s41746-020-00376-2

Gabel, V. P. (Ed.). (2017). *Artificial Vision*. https://doi.org/10.1007/978-3-319-41876-6

gaitan, Esteban Sánchez, & ampudia, margarita Malpartida. (n.d.). *revista médica sinergia*.

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow - Aurélien Géron - Google Libros* (3ra ed., Vol. 2). https://books.google.es/books?hl=es&lr=&id=X5ySEAAAQBAJ&oi=fnd&pg=PT10&dq=what+is+tensorflow&ots=yB5sog78wO&sig=d26PWG25dE-_PS9s2j8pAtuAclE#v=onepage&q=what%20is%20tensorflow&f=false

Hsu, T. C., Yang, H., Chung, Y. C., & Hsu, C. H. (2020). A Creative IOT agriculture platform for cloud computing. *Sustainable Computing: Informatics and Systems*, *28*. https://doi.org/10.1016/J.SUSCOM.2018.10.006

Janiesch, C., Zschech, P., & Heinrich, K. (n.d.). *Machine Learning and Deep Learning*. https://doi.org/10.1007/s12525-021-00475-2/Published

Javier, F., Rivera, V., Vargas, D. M., & Ruíz, M. Á. (n.d.). *Comparative Analysis of Interest Point Detectors Algorithms on Robotic Operative System*.

Juncosa, J., García, F., Campo, C., & González, T. (Eds.). (2022). *ANTROPOLOGÍAS HECHAS EN ECUADOR* (1ra ed., Vol. 1). Editorial Abya-Yala. https://asociacionlatinoamericanadeantropologia.net/portal/wp-content/uploads/2022/05/Antropologi%CC%81as-hechas-en-Ecuador-Tomo-I-11_mayo_2022-interactivo.pdf#page=448

Kelly, S. (2019). Python, PyGame, and Raspberry Pi Game Development. In *Python, PyGame, and Raspberry Pi Game Development*. Apress. https://doi.org/10.1007/978-1-4842-4533-0

Khemasuwan, D., Sorensen, J., & Colt, H. (2021). *Artificial intelligence in pulmonary medicine: computer vision, predictive model and COVID-19*. https://err.ersjournals.com/content/errev/29/157/200181.full.pdf

La siembra de chocho es más rentable | Revista Líderes. (n.d.). Retrieved May 3, 2022, from https://www.revistalideres.ec/lideres/siembra-chocho-produccion-chimborazo.html.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539

Li, Y., & Zhang, Y. (2020). Application research of computer vision technology in automation. *Proceedings - 2020 International Conference on Computer Information and Big Data Applications, CIBDA 2020*, 374–377. https://doi.org/10.1109/CIBDA50819.2020.00090

Lu, J., Hu, J., Zhao, G., Mei, F., & Zhang, C. (2017). An in-field automatic wheat disease diagnosis system. *Computers and Electronics in Agriculture*, *142*, 369–379. https://doi.org/10.1016/j.compag.2017.09.012

MAGAP. (2022). *INIAP investigó propiedades nutritivas del chocho, alternativa para una mejor alimentación – Ministerio de Agricultura y Ganadería*. https://www.agricultura.gob.ec/iniap-investigo-propiedades-nutritivas-del-chocho-alternativa-para-una-mejor-alimentacion/

Messaoud, S., Bouaafia, S., Maraoui, A., Ammari, A. C., Khriji, L., & Machhout, M. (2022). Deep convolutional neural networks-based Hardware–Software on-chip system for computer vision application. *Computers & Electrical Engineering*, *98*, 107671. https://doi.org/10.1016/J.COMPELECENG.2021.107671

Pajankar, A. (2021). Practical Linux with Raspberry Pi OS. In *Practical Linux with Raspberry Pi OS*. Apress. https://doi.org/10.1007/978-1-4842-6510-9

Quishpe Defaz, N. T. (2020, September). *"MODELO DE NICHO ECOLÓGICO PARA PREDECIR LA DISTRIBUCIÓN POTENCIAL DE ANTRACNOSIS (Colletotrichum acutatum) EN EL CULTIVO DE CHOCHO (Lupinus mutabilis Sweet)*. http://repositorio.utc.edu.ec/bitstream/27000/7086/1/PC-001051.pdf

Raspberry Pi Camera Module V2. (n.d.). Retrieved January 22, 2023, from https://www.pishop.us/product/raspberry-pi-camera-module-v2/.

Rees, C. A., & Ryder, H. F. (2022). Machine Learning for The Prediction of Ranked Applicants and Matriculants to an Internal Medicine Residency Program. *Teaching and Learning in Medicine*. https://doi.org/10.1080/10401334.2022.2059664

ROUHIAINEN, L. (2018). *INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO INTELIGENCIA ARTIFICIAL* (primera edición, Vol. 1). Editorial Planeta, S.A. www.planetadelibros.com

Rungta, K. (2018). *Cuda Update & tensorflow installation from scratch*. https://cmake.org/download/

Salvaris, M., Dean, D., & Tok, W. H. (2018). Deep learning with azure: Building and deploying artificial intelligence solutions on the microsoft AI platform. In *Deep Learning with Azure:*

Building and Deploying Artificial Intelligence Solutions on the Microsoft AI Platform. Apress Media LLC. https://doi.org/10.1007/978-1-4842-3679-6

Sánchez, J. (2017). *"CICLO DE VIDA DEL BARRENADOR DEL ÁPICE (DÍPTERA: Anthomyiidae) COMO PLAGA DEL CHOCHO, EN LA PROVINCIA DE COTOPAXI, PERIODO 2017*. 2017. http://repositorio.utc.edu.ec/bitstream/27000/4223/1/UTC-PC-000152.pdf

Sevillano-Bustos, F. J. (2015). *Equipo de clasificación de semillas basado en visión artificial. Implantación sobre hardware libre* [UNIVERSIDAD DE VALLADOLID]. https://uvadoc.uva.es/bitstream/handle/10324/17054/TFG-P-355.pdf?sequence=3&isAllowed=y

SOBRADO, E. A. (2020, August). *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot - ProQuest*. https://www.proquest.com/openview/1012aed86f88feec968e80c5af36dbb6/1?cbl=51922&diss=y&pq-origsite=gscholar

Tapia, M. E. (2015, July). *EL TARWI, LUPINO ANDINO*. http://fadvamerica.org/wp-content/uploads/2017/04/TARWI-espanol.pdf

Teachable Machine. (n.d.). Retrieved April 19, 2023, from https://teachablemachine.withgoogle.com/train.

Vaca M, J. L. (2021, July 16). *Ecografia de la Agrobiodiversidad del Chocho (Lupinus Multabilis Sweet) en el ecuador.* http://repositorio.utn.edu.ec/bitstream/123456789/11372/2/PG%20851%20TRABAJO%20GRADO.pdf

Watt, J., Borhani, R., & Katsaggelos, A. K. (2016). *Machine learning refined: foundations, algorithms, and applications* (1ra ed., Vol. 1).

Yepes, V., Martí, J. V., & García, J. (2020). Black hole algorithm for sustainable design of counterfort retaining walls. *Sustainability (Switzerland)*, *12*(7). https://doi.org/10.3390/SU12072767

Zhang, Q., Liu, Y., Gong, C., Chen, Y., & Yu, H. (2020). Applications of deep learning for dense scenes analysis in agriculture: A review. *Sensors (Switzerland)*, *20*(5). https://doi.org/10.3390/S20051520

Anexo

Anexo1. Etiquetado y detección de objetos con yoloV8

```
check gpu
!nvidia-smi
"""Instalar librerías"""
cd/content
import os
HOME= os.getcwd()
"""descargar  e instalar grouding DINO desde el repositorio"""
# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
!git clone https://github.com/IDEA-Research/GroundingDINO.git
# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}/GroundingDINO
!pip install -q -e .
!pip install supervision
"" Descargar peso de goudindino ya entrenados con  weinghts"""
# Commented out IPython magic to ensure Python compatibility.
!mkdir {HOME}/weights
# %cd {HOME}/weights
!wget -q https://github.com/IDEA-Research/GroundingDINO/releases/download/v0.1.0-
alpha/groundingdino_swint_ogc.pth
""" Ingresamos a la carpeta de goudingDINO
cd /content/GroundingDINO
"""funcionanmiento de goudingDINO
detección de objetos con groudingDINO
Cargamos weights.
# %cd {HOME}/GroundingDINO
from groundingdino.util.inference import load_model, load_image, predict, annotate
model =
```

```
load_model(f"{HOME}/GroundingDINO/groundingdino/config/GroundingDINO_SwinT_OGC.
py",   f"{HOME}/weights/groundingdino_swint_ogc.pth")
"""detección de objetos
importar librerías
import os
import supervision as sv
IMAGE_NAME = "/content/data/images/gnbn_03.jpg"
IMAGE_PATH = os.path.join(HOME, 'data', IMAGE_NAME)
TEXT_PROMPT = "good grain"
BOX_THRESHOLD = 0.45
TEX_THRESHOLD = 0.25

image_source, image = load_image(IMAGE_NAME)
boxes, logits, phrases = predict(
   model=model,
   image=image,
   caption=TEXT_PROMPT,
   box_threshold=BOX_THRESHOLD,
   text_threshold=TEX_THRESHOLD,
   device='cuda'
   )
annotated_frame = annotate(image_source=image_source,
                           boxes=boxes,
                           logits=logits,
                           phrases=phrases)
# %matplotlib inline
sv.plot_image(annotated_frame,(16,16))
"""etiquetado con groundingDINO"""
import local
locale.getpreferredencoding = lambda: "UTF-8"
!find /content/data/images -type d -name ".ipynb_checkpoints" -exec rm -rvf {} +
```

```
"""importar libreria de torch"""

#librerias
import os
from time import time
import cv2
import torch
#import torchvision.transforms as T
from PIL import Image
from GroundingDINO.groundingdino.util.inference import load_model, load_image, predict,
annotate
import GroundingDINO.groundingdino.datasets.transforms as T
#Device
DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
def read_images_from_folder(folder_path):
    images = []
    clases = []
    lista = os.listdir(folder_path)
    for lis in lista:
      img_path = os.path.join(folder_path, lis)
      img = cv2.imread(img_path)
      images.append(img)
      clases.append(os.path.splitext(lis)[0])

    return images, clases
def save_results(image, boxes, class_id, out_folder):
  #NORMALIZAR
  xc, yc, an, al = boxes[0][0], boxes[0][1], boxes[0][2], boxes[0][3]
  xc, yc, an, al = max(0, min(1, xc)), max(0, min(1, yc)), max(0, min(1, an)), max(0, min(1, al))
  list_info = [f"{class_id} {xc} {yc} {an} {al}"]
  time_now = str(time()).replace('.', '_')
```

```
  cv2.imwrite(f"{out_folder}/{time_now}.jpg", image)
  for info in list_info:
    with open(f"{out_folder}/{time_now}.txt", "a") as f:
      f.write(info)
def main():
  img_folder_path ='/content/data/images'
  out_folder_path = '/content/data/annotations'
  class_id = 0 # asignacion de clases
  save_results_flag = True
  images, classes = read_images_from_folder(img_folder_path)
  num_images = len(images)
  print(f"Imagenes: {num_images}")
  print(f'Nombres : {classes}')
  home = os.getcwd()

  #config path
  config_path = os.path.join(home,
"/content/GroundingDINO/groundingdino/config/GroundingDINO_SwinT_OGC.py")
  #checkpoint weights
  check_point_path = '/content/weights/groundingdino_swint_ogc.pth'
  #load model
  model = load_model(config_path, check_point_path)
  #prompt
  text_prompt =' good grain'
  box_threshold = 0.45
  text_threshold = 0.25
  for con in range(num_images):
    img = images[con]
    print("-----------//--------")
    print(f"Image: {classes[con]}")
    img_copy = img.copy()
```

```
    trasnform = T.Compose([
        T.RandomResize([800], max_size=1333),
        T.ToTensor(),
        T.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])

    img_source = Image.fromarray(img).convert("RGB")
    img_transform, _ = trasnform(img_source, None)
    boxes, logits, phrases = predict(
        model=model,
        image=img_transform,
        caption=text_prompt,
        box_threshold=box_threshold,
        text_threshold=text_threshold,
        device=DEVICE
    )
    if len(boxes) != 0:
      if save_results_flag:
        save_results(img_copy, boxes, class_id, out_folder_path)
    annotated_img = annotate(image_source=img,
                             boxes=boxes,
                             logits=logits,
                             phrases=phrases)
    out_frame = cv2.cvtColor(annotated_img, cv2.COLOR_RGB2BGR)
if __name__ == '__main__':
  main()
!zip -r "/content/data/annotations.zip" "/content/data/annotations"

"""borrado de imagenes"""
from shutil import rmtree
rmtree("/content/dataTrain/train/images")
```

```
!pip install ultralytics
cd /content/models
!wget https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt
"""train yolov8

#Train custom model
# Libraries
from ultralytics import YOLO
# Model
model = YOLO('/content/models/yolov8n.pt')
def main():
#Train
  model.train(data='/content/dataTrain/data.yaml', epochs = 40, batch = 64, imgsz = 640, device =
'cuda')
if __name__ == '__main__':
  main()
```

Anexo2. Código de login al Sistema embebido

Creación de una ventana de inicio para el acceso al sistema de clasificación del grano de chocho.

```
import tkinter as tk
from tkinter import ttk, messagebox
import sys
```

```
import subprocess

class AgriTechLogin:
    def __init__(self, root):
        self.root = root
        self.root.title("Sistema AgriTech - Clasificación de Grano")
        self.root.geometry("400x400")
        self.root.configure(bg="#1A4D2E")
        self.center_window()

        # Configuración de estilo
        self.style = ttk.Style()
        self.style.theme_use('clam')

        # Paleta de colores
        self.bg_color = "#1A4D2E"        # Verde oscuro
        self.accent_color = "#FF9F29"     # Naranja terroso
        self.danger_color = "#C40C0C"     # Rojo intenso
        self.text_color = "#F5F0ED"       # Blanco roto
        self.field_bg = "#E8F1F5"         # Azul claro

        self.font = ("Segoe UI", 10)
        self.title_font = ("Segoe UI", 16, "bold")

        self.create_widgets()

    def center_window(self):
        self.root.update_idletasks()
        width = self.root.winfo_width()
        height = self.root.winfo_height()
        x = (self.root.winfo_screenwidth() // 2) - (width // 2)
```

```
    y = (self.root.winfo_screenheight() // 2) - (height // 2)
    self.root.geometry(f'+{x}+{y}')

def create_widgets(self):
    main_frame = tk.Canvas(self.root, bg=self.bg_color, highlightthickness=0)
    main_frame.pack(fill=tk.BOTH, expand=True)

    header = tk.Frame(main_frame, bg=self.accent_color, height=100)
    header.pack(fill=tk.X)

    tk.Label(header, text="Acceso AgriTech", font=self.title_font,
            bg=self.accent_color, fg=self.bg_color).pack(pady=20)

    content_frame = tk.Frame(main_frame, bg=self.bg_color)
    content_frame.pack(pady=40, padx=40, fill=tk.BOTH)

    self.username = self.create_input_field(content_frame, "Usuario", "user")
    self.password = self.create_input_field(content_frame, "Contraseña", "lock", True)

    login_btn = ttk.Button(content_frame, text="Ingresar al Sistema",
                        command=self.validar_login, style="Accent.TButton")
    login_btn.pack(pady=20, fill=tk.X)

    exit_frame = tk.Frame(content_frame, bg=self.bg_color)
    exit_frame.pack(pady=10, fill=tk.X)

    ttk.Button(exit_frame, text="Salir", command=self.root.destroy,
            style="Danger.TButton").pack(fill=tk.X)

    self.style.configure("TEntry", font=self.font, padding=8,
                        fieldbackground=self.field_bg)
```

```
    self.style.configure("Accent.TButton", font=self.font,
                         background=self.accent_color, foreground=self.bg_color)
    self.style.configure("Danger.TButton", font=self.font,
                         background=self.danger_color, foreground=self.text_color)
    self.style.map("Accent.TButton", background=[('active', '#FFB740')])
    self.style.map("Danger.TButton", background=[('active', '#9D0000')])

def create_input_field(self, parent, label, icon_name, is_password=False):
    frame = tk.Frame(parent, bg=self.bg_color)
    frame.pack(fill=tk.X, pady=10)

    icon = tk.Label(frame, text=chr(0xE8B8) if icon_name == "user" else chr(0xE897),
                    font=("Segoe MDL2 Assets", 14), bg=self.bg_color, fg=self.text_color)
    icon.pack(side=tk.LEFT, padx=(0, 10))

    var = tk.StringVar()
    entry = ttk.Entry(frame, textvariable=var, show="*" if is_password else "")
    entry.pack(fill=tk.X, expand=True)

    entry.insert(0, label)
    entry.config(foreground='#7f8c8d')
    entry.bind("<FocusIn>", lambda e: self.on_entry_focus_in(entry, label))
    entry.bind("<FocusOut>", lambda e: self.on_entry_focus_out(entry, label))

    return var

def on_entry_focus_in(self, entry, placeholder):
    if entry.get() == placeholder:
        entry.delete(0, tk.END)
        entry.config(foreground='#000000')
```

```
    def on_entry_focus_out(self, entry, placeholder):
        if not entry.get():
            entry.insert(0, placeholder)
            entry.config(foreground='#7f8c8d')

    def validar_login(self):
        usuario = self.username.get()
        contraseña = self.password.get()

        if usuario == "davidft" and contraseña == "tesis2025":
            self.root.destroy()
            self.iniciar_sistema_clasificacion()
        else:
            messagebox.showerror("Error", "Credenciales incorrectas")

    def iniciar_sistema_clasificacion(self):
        try:
            subprocess.Popen([sys.executable, "clasificacion.py"])
        except Exception as e:
            messagebox.showerror("Error", f"Error al iniciar: {str(e)}")

if __name__ == "__main__":
    root = tk.Tk()
    app = AgriTechLogin(root)
    root.mainloop()
```

Anexo3. Codificación del sistema de clasificación

```
import sys
import cv2
import csv
```

```
from datetime import datetime
from PySide6.QtWidgets import (
    QApplication, QMainWindow, QLabel, QPushButton,
    QVBoxLayout, QHBoxLayout, QWidget, QGroupBox, QMessageBox
)
from PySide6.QtCore import QTimer, Qt
from PySide6.QtGui import QImage, QPixmap

class CameraApp(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Sistema de Clasificación de Granos")
        self.setGeometry(100, 100, 1000, 800)
        self.archivo_registros = "registros_granos.csv"

        self.cap = None
        self.timer = QTimer()

        self.contadores = {
            "bueno": 0,
            "malo": 0,
            "partido": 0,
            "haba": 0,
            "maiz": 0
        }

        self.init_ui()
        self.crear_archivo_registros()

    def init_ui(self):
        central_widget = QWidget()
```

```
        self.setCentralWidget(central_widget)
        main_layout = QVBoxLayout(central_widget)

        self.setStyleSheet("""
            QMainWindow { background-color: #2C3E50; }
            QLabel { color: #ECF0F1; font-size: 14px; }
            QPushButton {
                min-width: 80px;
                padding: 8px;
                border-radius: 5px;
                font-weight: bold;
            }
            QGroupBox {
                border: 2px solid #34495E;
                border-radius: 5px;
                margin-top: 10px;
                color: #BDC3C7;
            }
        """)

        self.video_label = QLabel("Cámara no activada")
        self.video_label.setAlignment(Qt.AlignCenter)
        self.video_label.setMinimumSize(640, 480)
        main_layout.addWidget(self.video_label)

        control_layout = QHBoxLayout()
        self.btn_iniciar = self.crear_boton("Iniciar Cámara", "#27AE60",
self.iniciar_camara)
        self.btn_detener = self.crear_boton("Detener Cámara", "#E74C3C",
self.detener_camara, False)
```

```
        self.btn_guardar = self.crear_boton("Guardar Registro", "#9B59B6",
self.guardar_registro)
        self.btn_cerrar = self.crear_boton("Cerrar Sistema", "#3498DB", self.close)

        control_layout.addWidget(self.btn_iniciar)
        control_layout.addWidget(self.btn_detener)
        control_layout.addWidget(self.btn_guardar)
        control_layout.addWidget(self.btn_cerrar)
        main_layout.addLayout(control_layout)

        contenedores_layout = QHBoxLayout()

        grupo_basicos = QGroupBox("Granos Básicos")
        grupo_basicos.setStyleSheet("QGroupBox { border-color: #16A085; }")
        layout_basicos = QHBoxLayout(grupo_basicos)
        layout_basicos.addWidget(self.crear_contador("bueno", "Bueno", "#2ECC71"))
        layout_basicos.addWidget(self.crear_contador("malo", "Malo", "#E74C3C"))
        layout_basicos.addWidget(self.crear_contador("partido", "Partido", "#F1C40F"))

        grupo_especiales = QGroupBox("Granos Especiales")
        grupo_especiales.setStyleSheet("QGroupBox { border-color: #8E44AD; }")
        layout_especiales = QHBoxLayout(grupo_especiales)
        layout_especiales.addWidget(self.crear_contador("haba", "Haba", "#9B59B6"))
        layout_especiales.addWidget(self.crear_contador("maiz", "Maíz", "#F39C12"))

        contenedores_layout.addWidget(grupo_basicos)
        contenedores_layout.addWidget(grupo_especiales)
        main_layout.addLayout(contenedores_layout)

        self.timer.timeout.connect(self.actualizar_frame)
```

```
def crear_boton(self, texto, color, funcion, habilitado=True):
    boton = QPushButton(texto)
    boton.setStyleSheet(f"""
        background-color: {color};
        color: white;
    """)
    boton.clicked.connect(funcion)
    boton.setEnabled(habilitado)
    return boton

def crear_contador(self, tipo, titulo, color):
    grupo = QGroupBox(titulo)
    layout = QVBoxLayout(grupo)

    grupo.setStyleSheet(f"""
        QGroupBox {{
            border-color: {color};
            font-size: 16px;
        }}
    """)

    lbl_contador = QLabel("0")
    lbl_contador.setStyleSheet("font-size: 24px; color: white;")
    lbl_contador.setAlignment(Qt.AlignCenter)
    layout.addWidget(lbl_contador)
```