



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA
DETECTAR PLAGAS EN LOS CULTIVOS DE ARROZ UTILIZANDO
UNA RASPBERRY PI Y OPENCV”**

AUTOR: Yostin Jahir Burgos Muñiz

TUTOR: Ph. D. Victor Huilcapi S.

Guayaquil, Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO
DE TITULACIÓN**

Yo, Yostin Jahir Burgos Muñiz, con documento de identificación N° 0929936938, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 07 de febrero del año 2024.

Atentamente,



Yostin Jahir Burgos Muñiz

0929936938

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DETITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo Yostin Jahir Burgos Muñiz, con documento de identificación No. 0953962313 e Iván Andrés Ventura Torres con documento de identificación No. 0929936938 expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del artículo académico: "diseño e implementación de un prototipo para detectar plagas en los cultivos de arroz utilizando una raspberry pi y Opencv" el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 08 de febrero del año 2024

Atentamente,



Yostin Jahir Burgos Muñiz
0929936938

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Victor Manuel Huilcapi Subia con documento de identificación N° 0917206294, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA DETECTAR PLAGAS EN LOS CULTIVOS DE ARROZ UTILIZANDO UNA RASPBERRY PI Y OPENCV** realizado por Yostin Jahir Burgos Muñiz N° 0929936938, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 07 de febrero del año 2024.

Atentamente,



Ph. D. Victor Manuel Huilcapi Subia

DEDICATORIA

A Dios por darme salud y sabiduría durante todo mi proceso académico, a mi madre y mis hermanos por darme el apoyo emocional en todo momento por estar a mi lado en cada paso que doy por darme fuerzas para tomar mis decisiones de salir adelante y por enseñarme valores de responsabilidad.

Yostin Burgos.

AGRADECIMIENTO

A Dios por haberme dado fuerzas en el arduo trabajo de toda mi carrera, por la salud y la bendición de llegar hasta este proceso de mi proyecto de vida.

A mi familia por estar en mis momentos difíciles durante todo el trayecto y que siempre me motivaron a seguir adelante.

A mis docentes por todo el conocimiento compartido durante todo mi proceso académico.

A mi tutor por todo el apoyo brindado en este trabajo de titulación y por el apoyo durante el proceso de aprendizaje.

ÍNDICE GENERAL

I INTRODUCCIÓN	1
II PLANTEAMIENTO DEL PROBLEMA	3
1.2 JUSTIFICACIÓN	4
III OBJETIVOS	5
3.1 Objetivo general	5
3.2 Objetivos específicos	5
IV MARCO TEÓRICO	6
4.1 ¿Qué son las plagas?	6
4.2.1 Desinfección	6
4.2.2 Desinsectación	6
4.2.3 Desratización	7
4.2.4 Fumigación	7
4.3 Afectación económica de las plagas en los cultivos	7
4.4 Control de plagas en los cultivos	8
4.4.1 Control químico	8
4.4.2 Control biológico	9
4.4.3 Control cultural	10

4.4.4 Control tecnológico.....	11
4.4.5 Manejo integrado de plagas.....	12
4.6 Plagas que afectan al los cultivo de arroz.....	13
4.4.6 “Mariposa blanca”	13
4.4.7 Efectos de la mariposa blanca	14
4.5 Cinta amarilla.....	14
4.5.1 Síntomas que causa la cinta amarilla	15
4.6 Como ayuda la inteligencia artificial en la agricultura.....	18
4.6.1 La inteligencia artificial y su papel en la adaptación al cambio climático	18
4.6.2 Pronóstico preciso de modelos de PREDICCIÓN.....	19
4.6.3 Agricultura de precisión y manejo adaptativo	19
4.6.4 Selección de cultivos resistentes al clima y plagas	19
4.6.5 Generalidades de la visión artificial.....	20
4.6.6 Inteligencia artificial	20
4.6.7 Ventajas de la inteligencia artificial	21
4.6.8 Proceso automatizado la inteligencia artificial.....	21
4.6.9 Mejorar las tareas creativas	21
4.7 Disminuir el error humano	21
4.8 Opencv.....	22

4.8.1 Técnicas de procesamiento de imágenes.....	22
4.8.2 Imagen digital	22
2.8.3 Imágenes fotográficas	23
2.8.4 Factores que afectan la imagen.....	23
4.8.5 Tecnología empleada en la detención de plagas.....	24
4.8.6 El monitoreo a distancia	24
2.8.7 Estrategia preventiva de la plaga en el cultivo	25
4.8.8 Modelo de detección yolo	25
4.8.8.1 YOLOv2:.....	26
4.8.8.2 YOLOv3:.....	26
4.8.8.3 YOLOv4:.....	26
4.8.8.4 YOLOv5.....	26
4.8.8.5 YOLOv6:.....	26
4.8.8.6 YOLOv7:.....	26
4.8.8.7 YOLOv8:.....	26
4.9 Scanner advanced ip scanner	27
• Escaneo de IP	27
• Recopilación de información:	27

• Interfaz de usuario:	27
• Funciones adicionales:.....	27
4.9.1 Aplicación realvnc viewer	28
Servidor VNC:.....	28
RealVNC Viewer:.....	28
Conexión Remota:.....	28
Control Remoto:.....	29
4.9.2 Tkinter	29
4.9.3 Almacenamiento de datos	29
4.9.5 Cámara ov5647	31
4.9.6 Sus características principales incluyen	31
4.9.8 Características principales de la raspberry pi 4 son:	33
4.9.9.1 Características de Python.....	35
4.9.9.2 Función del router	35
V MARCO METODOLÓGICO	37
5.1 Desarrollo del proyecto.....	37
5.2 Técnicas de investigación	38
5.3 Muestra	38
Diagrama de fases para la elaboración del prototipo.....	38

Diseño del prototipo	39
5.4 Desarrollo del prototipo para la detección de plagas en los cultivo de arroz.....	40
5.4.1 Instalación del sistema operativo	41
5.4.2 Proceso de conexión por medio del Router	43
5.4.3 Adquisición de imágenes	45
VI RESULTADOS.....	47
6.1 Creación del almacenamiento de datos con las muestras obtenidas de las plagas más comunes; cinta amarilla y mariposa blanca.	47
6.2 Entrenamiento del modelo en base al almacenamiento de datos creados.	53
4.3.1 Resultados del entrenamiento del modelo.....	55
6.3 Implementación y programación para el modelo de detención de plagas en la tarjeta raspberry pi 4.	59
6.4 Prueba final del prototipo funcionando.....	63
VII CRONOGRAMA.....	65
VIII PRESUPUESTO.....	66
IX CONCLUSIONES.....	67
X RECOMENDACIONES	68
XI REFERENCIAS BIBLIOGRÁFICAS	69
XII ANEXOS.....	74

ÍNDICE DE FIGURAS

FIGURA 1 Efecto económico potencial de las plagas y enfermedades.....	8
FIGURA 2 Control cultivo de arroz.....	9
FIGURA 3 Control biológico.....	10
FIGURA 4 Control cultural.....	11
FIGURA 5 Control tecnológico.....	12
FIGURA 6 Mariposa blanca.....	13
FIGURA 7 Cinta amarilla.....	15
FIGURA 8 Insecticida confidor.....	16
FIGURA 9 Herbicida Glifopac.....	17
FIGURA 10 Fungicida Bravo.....	17
FIGURA 11 Información de imágenes digitales.....	21
FIGURA 12 Open CV.....	22
FIGURA13 Gama de especialidades relacionadas al tratamiento de imágenes.....	24
FIGURA 14 Almacenamiento de datos.....	30
FIGURA 15 Cámara OV 5647.....	32

FIGURA 16 Raspberry Pi 4 placa base.....	33
FIGURA 17 Procesador cortex.....	34
FIGURA 18 Memoria RAM 8gb.....	34
FIGURA 19 Gráfico 4k.....	34
FIGURA 20 Python.....	35
FIGURA 21 Función de Router.....	36
FIGURA 22 Diagrama de fases para la elaboración del prototipo.....	39
FIGURA 23 Diseño del prototipo.....	39
FIGURA 24 Conexión de la cámara al raspberry pi 4 en caset con disipador de calor..	40
FIGURA 25 Router para conexión remota hacia la raspberry pi 4.....	41
FIGURA 26 Sistema operativo Raspbian instalado.....	43
FIGURA 27 Dirección IP de la Raspberry pi 4.....	44
FIGURA 28 Propiedades del dispositivo raspberry pi.....	44
FIGURA 29 Pestaña de autenticación de usuario y contraseña.....	45
FIGURA 30 Conexión remota a red Wi-fi.....	45
FIGURA 31 Recolección de muestras de las plagas.....	46
FIGURA 32 Comando para ejecutar labelme.....	47
FIGURA 33 Carpeta de imágenes seleccionadas.....	48

FIGURA 34 Muestra afectada cinta amarilla.....	48
FIGURA 35 Etiquetado de la muestra afectada.....	49
FIGURA 36 Etiquetado de la muestra.....	49
FIGURA 37 Etiquetado de cada imagen cinta amarilla.....	50
FIGURA 38 Etiquetado de cada imagen mariposa blanca.....	50
FIGURA 39 Referencia de los puntos de las imágenes.....	51
FIGURA 40 Comando de conversión.....	51
FIGURA 41 Conversión de las muestras.....	52
FIGURA 42 Conversión finalizada.....	52
FIGURA 43 Almacenamiento de datos.....	52
FIGURA 44 Comando para entrenar el modelo.....	53
FIGURA 45 Épocas de entrenamiento.....	54
FIGURA 46 Entrenamientos realizados.....	55
FIGURA 47 Resultado del entrenamiento del modelo.....	55
FIGURA 48 Resultado de detección de la mariposa blanca.....	56
FIGURA 49 Detección de mariposa blanca.....	56
FIGURA 50 Entrenamiento del modelo cinta amarilla y mariposa blanca.....	57
FIGURA 51 Entrenamiento del modelo mariposa blanca.....	57

FIGURA 52	Resultado del entrenamiento del modelo.....	58
FIGURA 53	Archivo ModeloPlagas.....	59
FIGURA 54	Modelo instalado en la raspberry pi 4	59
FIGURA 55	Código importación de librería.....	60
FIGURA 56	Código de interfaz.....	60
FIGURA 57	Código función iniciar.....	61
FIGURA 58	Código de selección del modelo entrenado.....	61
FIGURA 59	Función visualizar.....	62
FIGURA 60	Código para desactivar la interfaz.....	62
FIGURA 61	Detención de plaga mariposa blanca.....	63
FIGURA 62	Captura detención de plaga mariposa blanca.....	63
FIGURA 63	Detención de plaga cinta amarilla.....,	64
FIGURA 64	Captura detención de plaga cinta amarilla.....	64

ÍNDICE DE TABLAS

TABLA 1 Cronograma de actividades.....65

TABLA 2 Presupuesto para la implementación del proyecto.....66

RESUMEN

Las principales complicaciones que presentan los cultivos de arroz y que necesitan ser eliminados rápidamente son las plagas, las cuales hacen contacto directo con ellas provocando daños tanto a la planta como al grano, teniendo en cuenta que en ocasiones puede ser combatido si se lo detecta inmediatamente. El proyecto se centra en desarrollar un modelo inicial que pueda identificar por medio de una cámara que se implementará en la raspberry pi 4 contará con almacenamiento de datos programado para reconocer plagas como la mariposa blanca y la cinta amarilla. Este proyecto trata sobre detectar a tiempo las plagas que afectan a los cultivos, siendo Ecuador unos de los países con mayor aporte económico en la agricultura individualmente en los cultivos de arroz por esa razón es importante enfocarse en facilitar medidas para la detención temprana de plagas.

Es evidente que haya factores que intervengan a la detección de plagas en el cultivo obteniendo un porcentaje de detención correcta e incorrecta, sin embargo, el algoritmo empleado en Python ayudará a reconocer su objetivo mediante herramientas como Open CV y Ultralytics, dando con la detección de la mariposa blanca y la cinta amarilla.

Palabras claves: Cultivos, Plagas, detectar, afectan, enfocarse, resultados, porcentaje

ABSTRACT

One of the main problems that arise for rice crops and that must be eliminated quickly are pests, which make direct contact with them causing damage to both the plant and the grain taking into account that sometimes it can be fought if it is detected immediately. This project focuses on the design of a prototype that detects by means of a camera implemented in the raspberry pi 4 with a data storage programmed to recognize pests such as the rice bride and yellow tape. This project is about early detection of pests that affect crops, being Ecuador one of the countries with greater economic contribution in agriculture specifically in rice crops for that reason it is important to focus on facilitating measures for early detection of pests.

It is evident that there are factors that intervene to the detection of pests in the crop obtaining a percentage of correct and incorrect detention, however, the algorithm used in Python will help to recognize its target through tools such as Open CV and Ultralytics, giving the detection of the white butterfly and the yellow ribbon.

Key words: Crop, Pests, detect, affect, focus, results, percentage, percentage.

I INTRODUCCIÓN

En el presente proyecto se usará la inteligencia artificial enfocándose en el diseño de un prototipo que ayude al reconocimiento de plagas que afectan a los cultivos, para la implementación de este trabajo se necesitara la adquisición de imágenes de los tipos de plagas que se pretende detectar por medio de la cámara de la raspberry pi 4, las cuales van a capturarse imágenes que contengan presencia temprana de la “novia del arroz” y “cinta amarilla”, los cultivos de arroz al ser constantemente amenazados por estas plagas y por la cantidad de dinero que generalmente se pierde y se gasta para reparar daños a las partes afectadas se considera esta una opción para realizar este proyecto.

Sin embargo, los agricultores creen más apto el uso de productos como los insecticidas, fungicidas y herbicidas para sus cultivos, lo cual es un proceso más costoso y que en ocasiones suele ser dañino someter al arroz con productos químicos.

El capítulo I se manifestará los antecedentes para abordar las problemáticas que enfrentan los cultivos sobre las plagas y dado los objetivos que se enfocan en implementar medidas para la detención de este.

El capítulo II se va a desarrollar el marco teórico explica las características de estas plagas en cómo afecta a los cultivos basándose en el uso de la tecnología en su implementación y como ayudaría la inteligencia artificial para la detención temprana de plagas.

El capítulo III se realizará el diseño del prototipo para la detención de la novia del arroz y la cinta amarilla, en el cual se especifican lo que se va a utilizar y las características de lo que se realizará.

El capítulo IV mostrará los resultados obtenidos del prototipo funcionando, se obtendrá la detención de las plagas como la cinta amarilla y la mariposa blanca.

Finalmente se muestran las conclusiones y recomendaciones explicando como resultados el prototipo terminado.

II PLANTEAMIENTO DEL PROBLEMA

En República del Ecuador. Instituto Nacional de investigaciones agropecuarias registra distintos tipos de plagas que dañan nuestra producción tales como, la mosca minadora del arroz (*Hydrellia*) ataca y perfora la lámina foliar (Hojas), se agrupan o se fusionan causando así que esa parte de la hoja se necrosifica y finalmente se quiebra, se expanden en las cicatrices y las áreas afectadas provocando así una decoloración amarilla en la punta de la hoja, esta plaga mide aproximadamente desde 0.5 a 10.0 cm. de longitud y durante todo su ciclo, pueden causar daños graves a la planta.

Estas plagas representan dificultades para los cultivos, al trabajarse de manera rápida antes de que ataque por completo la producción y se pierda la cosecha, existen diversas alternativas para combatirlas entre ellos procesos costosos y a lo largo del mantenimiento para cuidar el arroz hay productos dañinos para la salud en que se utilizan con el fin de acabar con las plagas.

La agricultura es un factor principal para la economía ecuatoriana, por esta razón el cultivo de arroz es uno de los sectores agrícolas más importantes del país. Sin embargo, están constantemente amenazados por plagas que pueden afectar significativamente la producción y la calidad del arroz.

Daule es uno de los cantones que posee una estructura laboral, productiva y bastante participativa en cultivo de arroz, por otra parte, este sector se ve afectado particularmente por diferentes plagas muy comunes en este caso entran la mariposa blanca y la cinta amarilla que afecta rápidamente a nuestra cosecha si no se detecta

desde el inicio causando una disminución en el volumen de producción, cada vez esto se vuelve más particular que se pierda grandes cantidades de cultivos por no detectar las plagas a tiempo, esto comúnmente implica mayor cantidad de dinero y tiempo para solucionar estos problemas.

1.2 JUSTIFICACIÓN

La detención temprana de plagas en los cultivos de arroz es crucial para minimizar los daños y pérdidas en la producción agrícola, se tiene presente que estos procesos son laboriosos, se requieren de expertos para su realización así invirtiendo mucho más tiempo para ello, lo que representa un desafío para los agricultores.

El presente proyecto pretende ayudar a los agricultores con el diseño de un prototipo que detecte plagas en los cultivos de arroz, utilizando tecnologías innovadoras como visión por computadora, programación y aprendizaje automático. Brindado a los agricultores la oportunidad de tomar medidas de prevención a tiempo, para así evitar grandes pérdidas y tener una mejor producción de sus cultivos.

III OBJETIVOS

3.1 Objetivo general

Diseñar e implementar un prototipo que ayude a la detección temprana de plagas para los cultivos de arroz.

3.2 Objetivos específicos

- Tomar muestras de imágenes de las plagas más comunes en el cultivo del arroz; mariposa blanca, cinta amarilla, para crear conjunto de datos.
- Entrenar un modelo basado en visión por computadora para la detención de plagas con base al conjunto de datos creado.
- Implementar el modelo de detención de plagas en un prototipo usando la tarjeta Raspberry Pi y Open CV.

IV MARCO TEÓRICO

4.1 ¿Qué son las plagas?

Las plagas más destructivas son altamente adaptables, es decir, adaptadas a varios escenarios y condiciones ecológicas del mundo. Las plagas logran activarse en épocas específicas o durante todos los días del año. Las poblaciones de insectos se consideran plagas cuando reducen la calidad o cantidad de los alimentos. Sus tasas de reproducción varían, pero en su totalidad se reproducen con rapidez y puede causar daños a los árboles en operaciones agrícolas o forestales. (Juarez, 2023)

4.2 Tipos de procesos del control de plagas

4.2.1 Desinfección

La desinfección implica la utilización de métodos químicos o físicos para limpiar superficies y espacios, con el objetivo de eliminar microorganismos patógenos. Es esencial en entornos con afluencia humana o animal, como escuelas, hospitales, granjas, industrias alimentarias y vehículos de transporte. (IEOFB, 2020)

4.2.2 Desinsectación

La desinsectación se refiere al manejo y erradicación de insectos de diversa índole. Entre las plagas mencionadas previamente se encuentran algunos de estos insectos. (IEOFB, 2020)

4.2.3 Desratización

La desratización busca erradicar roedores utilizando dos enfoques, a través de procedimientos que eviten la propagación de plagas y mediante métodos ofensivos que involucran químicos y biológicos para combatir las poblaciones. (IEOFB, 2020)

4.2.4 Fumigación

La fumigación, mediante el uso de fumigantes, constituye un método altamente eficaz para eliminar invasiones de insectos, aves o microbios. Estos agentes químicos se encuentran en estado gaseoso a temperatura y presión normales, y se aplican principalmente en la industria alimentaria, tratamientos ambientales, tratamiento de la madera y control de plagas. (IEOFB, 2020)

4.3 Afectación económica de las plagas en los cultivos

Las repercusiones económicas de las plagas pueden llegar a ser complicadas y la atención del proyecto está dirigida a crear un prototipo capaz de reconocer. (como se ilustra en la Figura 1), dando lugar a una variedad de posibles impactos. En situaciones particulares, la magnitud real del impacto económico dependerá de la clase específica de enfermedad o plaga, si bien la complejidad a menudo dificulta una medición precisa de dicho impacto. (FAO, s.f.)

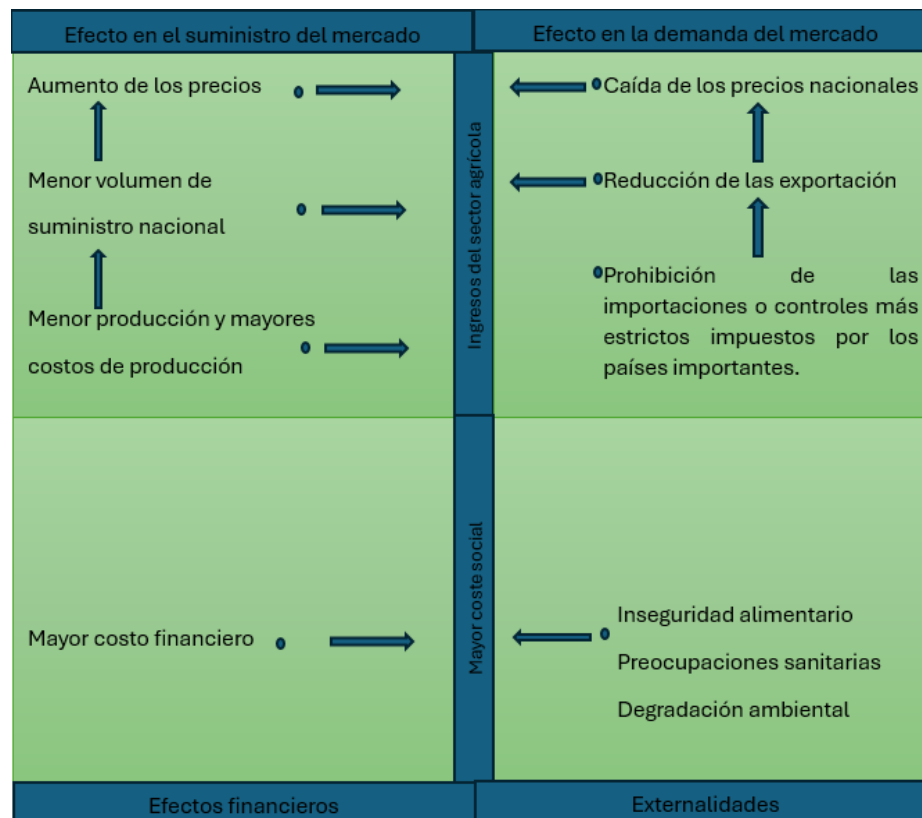


Figura 1 efecto económico potencial de las plagas y enfermedades (FAO, s.f.)

4.4 Control de plagas en los cultivos

Hay muchos tipos de agentes de control de plagas, pero cada agente de control de plagas tiene sus ventajas y sus desventajas correspondientes. Por ejemplo, los pesticidas son productos químicos naturales o sintéticos que tienen propiedades específicas y pueden producir efectos tóxicos en ciertos organismos en determinadas dosis. (Juarez, 2023)

4.4.1 Control químico

La gestión química, ver en la figura 2, es una opción relativamente eficaz y rápida considerando las prácticas de manejo, aplicación y almacenamiento. qué se

requiere para frenar una intoxicación o evitar resultados negativos a los cultivos, entre las siguientes medidas que se pueden realizar un control químico son:

- Aspersión terrestre.
- Inyección al tronco.
- Aplicación al suelo.



Figura 2 control cultivo de arroz (Alamy, 2012)

4.4.2 Control biológico

El control para los cultivos es una táctica sostenible para reducir la población utiliza organismos vivos de plagas sin dañar el ambiente, ni causar daños a la salud humana. Algunas relevantes sobre el control biológico en los cultivos incluyen lo siguiente:

- Agentes del control.
- Estrategias de implementación.
- Reducción de usos de pesticidas.

El control biológico ver figura 3 no solo ofrece distintas alternativas más sostenible y considerada con el medio ambiente, sino que también logra contribuir a la

seguridad alimentaria al mantener la salud de los cultivos y reducir las pérdidas debido a las plagas. (Juarez, 2023)



Figura 3 control biológico (Corporación AMA, 2023)

4.4.3 Control cultural

Para el control cultural ver figura 4 se refiere a prácticas agrícolas que buscan prevenir o reducir las plagas y enfermedades de manera natural a través de técnicas específicas de manejo del cultivo. Algunas estrategias de control cultural en el cultivo incluyen:

- Poda Fertilización.
- Control del agua.
- Manejo de residuos y malezas. Derribo (Juarez, 2023)



Figura 4 control cultural del cultivo de arroz (Engormix, 2015)

4.4.4 Control tecnológico

El uso de tecnología es una medida avanzada que se utiliza para monitorear, gestionar y optimizar el cultivo, incluyendo métodos de detección y respuesta rápida ante las plagas, enfermedades y otros problemas que puedan afectar la producción agrícola, algunas de las tecnologías utilizadas en el control tecnológico de cultivos son:

- Aspersión aérea ver figura 5.
- Sensores remotos.
- Aplicaciones de análisis de datos. (Juarez, 2023)



Figura 5 control tecnológico en el cultivo de arroz. (red agrícola, 2019)

4.4.5 Manejo integrado de plagas

Es esencial emplear el manejo integrado de las plagas para prevenir inconvenientes derivados del control químico. Este enfoque implica Emplear todos los recursos necesarios a través de procedimientos operativos estándar con el objetivo de minimizar los riesgos asociados a la presencia de plagas, a diferencia de los métodos tradicionales de control de plagas.

El objetivo inicial es proteger tanto como sea posible. Por tanto, la base principal del MIP es la introducción, adaptación y viabilidad económica de cualquier especie o variedad. Existe un sistema denominado Sistema de Gestión Ecológica, en el que el conocimiento básico de una determinada placa nos permitirá proponer, programar, organizar, integrar y predecir las técnicas de combate disponibles. Cuando las plagas van más allá del marco de prevención, el marco debe entenderse como complementario a los controles naturales y debe

usarse para predecir el impacto de cada tecnología que constituye los controles biológicos naturales. (Juarez, 2023)

4.6 Plagas que afectan al los cultivo de arroz

El cultivo de arroz puede enfrentar varias plagas que pueden reducir la calidad y producción de la cosecha pueden verse afectadas si no se gestionan de manera adecuada. (Plantix, 2023)

Algunas de las plagas más comunes que ocasionan daños al cultivo de arroz incluyen:

4.4.6 “Mariposa blanca”

Rupela Albinella la novia del arroz o mariposa blanca se muestra en la figura 6 le cae a la hoja, tallo y espiga de *arroz* ocasionando debilidad, amarillamiento y se marchite el cultivo (Plantix, 2023).



Figura 6 mariposa blanca (Engormix, 2018)

4.4.7 Efectos de la mariposa blanca

Las complicaciones surgen cuando las pequeñas larvas, al emerger de los huevos, perforan el tallo y los entrenudos, formando galerías que interfieren con el transporte de nutrientes hacia la parte superior de la planta. Este proceso puede resultar en tornarse un color amarilloso de las hojas inferiores o el daño conocido como panícula blanca, dependiendo de la edad de la planta. Si la infestación de la mariposa sucede después del inicio de la paniculación, aunque no influye en la translocación de nutrientes además de que no puede observarse la panícula blanca ni granos llenos. Aunque en general lo afectado suele tener poca relevancia económica, en algunas áreas arroceras puede manifestarse en poblaciones elevadas en determinadas circunstancias. (AgroBase, s.f.)

4.5 Cinta amarilla

Cinta amarilla afecta del 40 al 60% los rendimientos de arroz, después de que esta se infecte se forman manchas oscuras en las venas de las hojas apareciendo una coloración de franjas amarillas se evidencia en la figura 7 en el centro lo que hace que esta infección se inhibe en la planta y su producción se reduzca (Plantix, 2023).



Figura 7 Cinta Amarilla (Plantix, 2023)

4.5.1 Síntomas que causa la cinta amarilla

Las hojas más jóvenes muestran manchas amarillas a verdes dos semanas después de la infección, que se extienden en paralelo a las venas. En el centro de estas áreas amarillas, aparecen manchas oscuras. En las hojas más antiguas, se presenta una tonalidad amarillo naranja, afectando el crecimiento de las plantas e impactando negativamente en su producción. (Plantix, s.f.)

4.5.2 Tipos de químicos que se utilizan para el control de plagas

Las metodologías empleadas para el control de las plagas por procesos de sustancias químico se consideran eficaces, las cuales pueden llegar a ser nocivas para el medio ambiente, personas y para el cultivo directamente, están cada vez más restringidas. La selección del método adecuado dependerá del tipo de plaga, el contexto en el que se manifiesta y otros factores relevantes. (ADEPAP, 2023)

Los métodos químicos más usuales para manejar las plagas en áreas urbanas incluyen los insecticidas, herbicidas y fumigantes.

4.5.3 Insecticidas

Los insecticidas, representados en la figura 8, son sustancias químicas empleadas para gestionar o eliminar insectos que puedan portar enfermedades. Es crucial tener en consideración que ciertos insecticidas pueden ser perjudiciales para la salud. (Pública, 2020)



Figura 8 insecticida Confidor (Agrofortaleza 2023)

4.5.4 Herbicidas

El herbicida es una sustancia química que se utiliza para impedir o detener el crecimiento de las malezas o plantas no deseadas, comúnmente denominadas hierbas malas afectando áreas que serán cultivadas o ya han sido cultivadas. (Agrotterra, 2021)



Figura 9 Herbicida Glifopac (Agrizon 2023)

4.5.5 Fungicidas

Los fungicidas, ver la figura 9, son sustancias tóxicas utilizadas para prevenir el crecimiento o para poder eliminar los mohos y hongos que perjudican las plantas o animales. Aunque los fungicidas pueden ser eficaces, su uso excesivo puede provocar daños fisiológicos en las plantas. Dado que son productos químicos, es esencial emplearlos con cautela para evadir cualquier riesgo para la salud de la persona, animal y/o el medio ambiente. (Tecnosoluciones, 2019)



Figura 10 Fungicida Bravo (Agrizon 2023)

4.5.6 Resistencia de las plagas a los productos químicos

La resistencia es evidenciada como una muestra de plagas ha desarrollado capacidad genética de sobrevivir a una cantidad de insecticida que sería mortal para la población original de la plaga. Puede variar en grado, desde tolerar una dosis más alta hasta alcanzar una insensibilidad práctica al producto. (Higieneambiental, 2016)

La evaluación del nivel de resistencia implica contrastar a la población de su sensibilidad tenaz con la de una población débil de referencia utilizando la concentración o dosis es de 50 (DL50), que es la cantidad necesaria para eliminar al 50% de la población. (Higieneambiental, 2016)

El factor de resistencia (FR50) se calcula dividiendo la DL50 de la población resistente entre DL50 de la población débil. Un FR50 de 2 indica que se debe duplicar la dosis para lograr el mismo efecto en la población resistente. Si el FR50 es 100, se requiere multiplicar la dosis por 100. Es crucial señalar que la presencia de resistencia no siempre implica que exista en el entorno natural, siempre y cuando las dosis toleradas no superen la aplicación de productos en el campo. (Higieneambiental, 2016)

4.6 Como ayuda la inteligencia artificial en la agricultura

4.6.1 La inteligencia artificial y su papel en la adaptación al cambio climático

La inteligencia artificial se percibe como una herramienta esencial para apoyar a los agricultores en adaptarse y reducir los impactos del cambio climático. Al hacer uso de la inteligencia artificial, los agricultores pueden tomar decisiones

fundamentadas en datos y anticiparse a las consecuencias del cambio climático. (Kubiliute, 2023)

4.6.2 Pronóstico preciso de modelos de PREDICCIÓN

La inteligencia artificial tiene la capacidad de examinar extensos vínculos de datos históricos y del tiempo real, que abarcan información climática, del suelo y de los cultivos, con el fin de generar predicciones precisas. Estos pronósticos posibilitan al agricultor planifiquen y adapten sus actividades agrícolas de la siembra del cultivo hasta la cosecha. (Kubiliute, 2023)

4.6.3 Agricultura de precisión y manejo adaptativo

La tecnología de agricultura e inteligencia artificial da precisión, que incluyen sensores, drones y sistemas de vigilancia, además de que ayudan a obtener información detallada sobre las condiciones de la humedad del suelo y el cuidado de las plantas, entre otros componentes relevantes. Los datos se analizan mediante algoritmos de inteligencia artificial para generar recomendaciones específicas y ajustables para la gestión agrícola. De este modo, los agricultores pueden afinar con precisión las aplicaciones de riego, fertilizantes, pesticidas y otras prácticas agrícolas, buscando optimizar la eficiencia y reducir el impacto ambiental. (Kubiliute, 2023)

4.6.4 Selección de cultivos resistentes al clima y plagas

La inteligencia artificial puede ser de utilidad en la selección de variedades de cultivos que sean más resistentes al clima y capaces de adaptarse a nuevas

condiciones. Al examinar datos genéticos y emplear algoritmos de aprendizaje automático, la IA puede identificar características genéticas que confieren resistencia a desafíos climáticos como la sequía o temperaturas extremas, así como factores como plagas que causan daño directo a la producción de cultivos. Esto brinda a los agricultores la capacidad de tomar decisiones fundamentadas al elegir los cultivos más apropiados para sus condiciones específicas. (Kubiliute, 2023)

4.6.5 Generalidades de la visión artificial

4.6.6 Inteligencia artificial

La inteligencia artificial IA es un campo que ha transformado disciplinas, economía e industrias, se basa mediante procesos de aplicación de algoritmos creado de forma dinámica por computadora para que estas actúen y extraigan información por medio de imágenes digitales ver figura 10, tomar medidas o realizar recomendaciones mediante la información que recibe, así permite a los ordenadores pensar, observar y comprender, en conclusión, la IA es una tecnología que da nuevas soluciones innovadoras. (NetApp, 2023)



Figura 11 información de imágenes digitales (Exevi, 2022)

4.6.7 Ventajas de la inteligencia artificial

Las principales ventajas de la IA aplicadas:

4.6.8 Proceso automatizado la inteligencia artificial

Habilita a los prototipos a realizar de forma inmediata tareas repetitivas, rutinarias y de optimización de técnicas sin la necesidad de la intervención humana. (NexusIntegra, s.f.)

4.6.9 Mejorar las tareas creativas

La IA da paso de obtener a las personas labores repetitivas y rutinarias, posibilitándoles emplear más tiempo en el desarrollo de habilidades creativas. (NexusIntegra, s.f.)

4.7 Disminuir el error humano

La inteligencia artificial reduce las fallas debido a las limitaciones humanas.

En algunos vínculos de producción se recurre a la inteligencia artificial para detectar se debe utilizar sensores infrarrojos para pequeñas grietas o fallas en piezas que no pueden ser detectadas por el ojo humano. Reducir el tiempo dedicado al análisis de datos permite analizar y explotar los datos que van a ser derivados de la producción se pueden llegar a efectuar en tiempo real. (NexusIntegra, s.f.)

4.8 Opencv

Este sistema da nueva oportunidad que ayuda a potenciar y explorar el avance de la tecnología, es una biblioteca libre Open CV observe figura 11 permite identificar objetos, clasificar acciones, reconocer escenarios, extraer modelos en 3D, realizar tracking de movimientos, OpenCV tiene una licencia de BSD que permite utilizar y modifica el código, su virtud es multiplataforma puede ejecutar diferentes sistemas operativos, también se puede utilizar para distintos lenguajes de programación (Rodríguez, 2021).

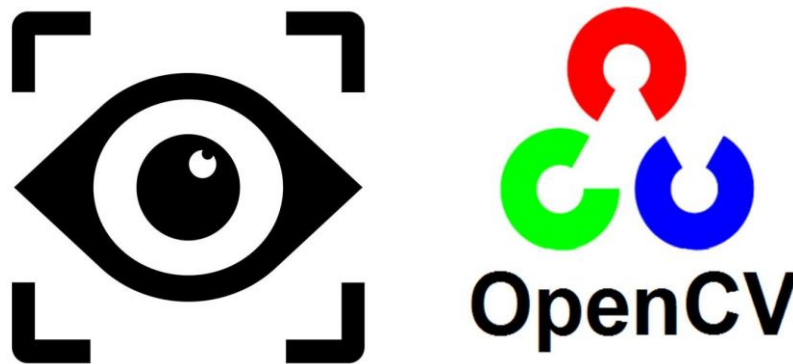


Figura 12 Open CV (Analytics Vidhya, 2021)

4.8.1 Técnicas de procesamiento de imágenes

4.8.2 Imagen digital

Una representación digital de una imagen se define como una función discreta y finita en un espacio bidimensional (x, y) , donde los elementos se llaman píxeles. Cada píxel, siendo la unidad más pequeña, tiene un valor funcional discreto y finito. La generación de una imagen digital puede ocurrir de dos maneras: mediante la interacción de una fuente de energía con el entorno físico o a partir de datos o

imágenes preexistentes. En el primer caso, implica una fuente de energía interactuando con la escena física, un sensor que registra la respuesta de la escena y un digitalizador que convierte la señal analógica del sensor en una señal digital. (González, 2021)

2.8.3 Imágenes fotográficas

Específicamente, una imagen fotográfica se logra proyectando objetos del mundo real en un plano, capturando esta proyección mediante el registro de la interacción de la radiación electromagnética visible con los objetos fotografiados. En esta proyección, las imágenes de los objetos de la escena pueden superponerse, lo que resulta en la pérdida de la dimensión asociada a la profundidad. (González, 2021)

2.8.4 Factores que afectan la imagen

La calidad de la imagen se evalúa considerando factores clave como la resolución, el color, el tono y la apariencia general. En imágenes a color, en escala de grises y en algunas monocromáticas, la reproducción precisa del color y del tono se vuelve crucial para la calidad, complementando el "detalle" proporcionado por la resolución. Al evaluar la apariencia del color y del tono, el objetivo es determinar cómo la imagen comunica la misma apariencia mientras el color y el tono varían en comparación con el documento original (o el intermedio utilizado). La evaluación del tono y del color puede ser subjetiva y susceptible a variaciones según el entorno de visualización y las características de los monitores e impresoras. (CornellUniversity, 2003)

Para el uso del computador para tratar imágenes se ha entendido en los últimos años, y se diferencian las siguientes áreas, como se puede ver en la imagen 12

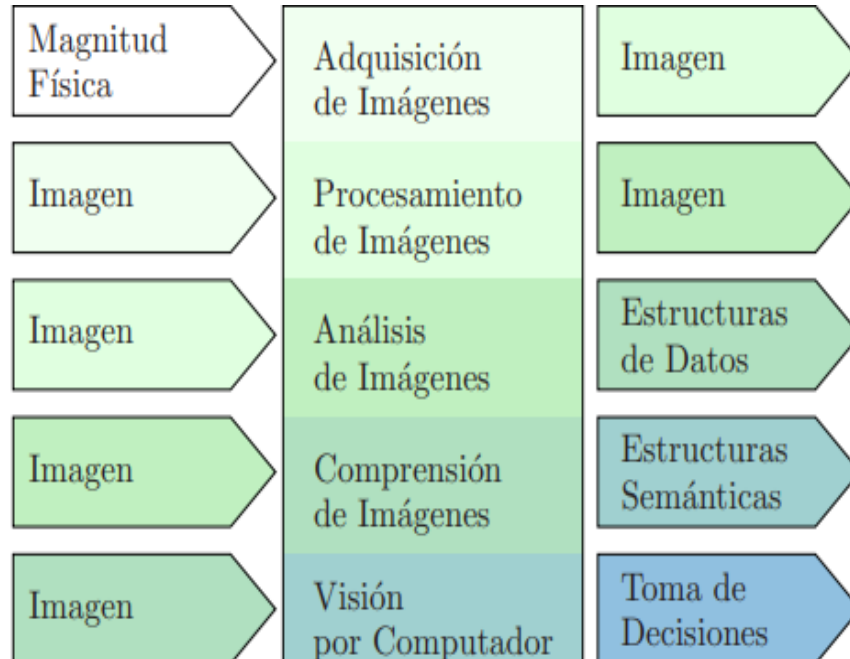


Figura 13 Gama de especialidades relacionadas con el tratamiento de imágenes (Alvarado, 2012)

4.8.5 Tecnología empleada en la detención de plagas

4.8.6 El monitoreo a distancia

Una vez que se detecta e identifica la plaga en el cultivo y gracias al avance de internet en la agricultura, los datos se cargan en la nube digital, y el propietario de la plantación recibe una notificación instantánea en su dispositivo móvil sobre lo que está ocurriendo en su finca. Un aspecto relevante es que los sensores son de baja potencia y operan con baterías simples, lo que significa que el sistema tiene un costo competitivo en comparación con otros métodos de control, como la vigilancia

manual, proporcionando información confiable sobre la población de plagas en tiempo real. (Martín, 2020)

Esta herramienta asombrosa también permite el control a nivel regional de plagas que representan problemas para los cultivos, sirviendo como un sistema de alerta en caso de plagas provenientes de otras áreas, permitiendo a los productores estar prevenidos ante posibles futuros ataques. (Martín, 2020)

2.8.7 Estrategia preventiva de la plaga en el cultivo

Establece cuándo es el momento adecuado para implementar medidas y no utilizar fitosanitarios según fechas específicas del calendario, como se hacía previamente para garantizar el control sin certeza. Sin duda, esto implica un ahorro significativo de tiempo y dinero, ya que la aplicación se realiza únicamente cuando es necesario. (Martín, 2020)

4.8.8 Modelo de detección yolo

Un destacado sistema de detección y segmentación de objetos en imágenes, basado en aprendizaje profundo y visión computarizada, presenta avances significativos, proporcionando un rendimiento sin igual en velocidad y precisión. Gracias a su diseño aerodinámico, se adapta con facilidad a diversas aplicaciones y puede integrarse sin dificultad en diversas plataformas de hardware, desde dispositivos de borde hasta APIs en la nube. (AyushExel, 2023)

Yolo tiene varios modelos desde YOLOv2 del 2016 hasta YOLOv8:

4.8.8.1 YOLOv2: Mejoró en 2016 su predecesor al incorporar normalización por lotes, cajas de anclaje y grupos de dimensiones. (AyushExel, 2023)

4.8.8.2 YOLOv3: Lanzado en 2018, llevó a cabo mejoras adicionales mediante una red troncal más eficiente, anclajes múltiples y agrupación de pirámides espaciales.

4.8.8.3 YOLOv4: Presentado en 2020, Presentó novedades tales como la expansión del conjunto de datos Mosaico, un innovador cabezal de detección sin anclajes y una función de pérdida actualizada. (AyushExel, 2023)

4.8.8.4 YOLOv5: Amplió el rendimiento y funciones con Ajuste de hiperparámetros, monitoreo integrado de experimentos y exportación automatizada a formatos populares. (AyushExel, 2023)

4.8.8.5 YOLOv6: De código abierto en 2022, es utilizado por Meituan en sus robots autónomos de reparto. (AyushExel, 2023)

4.8.8.6 YOLOv7: Incluyó nuevas tareas, optimización de hiperparámetros, experimentos con seguimiento integrado y exportación automática a diferentes formatos. COCO de puntos clave. (AyushExel, 2023)

4.8.8.7 YOLOv8: La última versión de Ultralytics, eleva el estándar con nuevas funciones y mejoras, brindando rendimiento, flexibilidad y eficacia mejorados en tareas de visión por IA, como detección, segmentación, estimación de pose, seguimiento y clasificación. Su versatilidad permite su aplicación en diversas áreas y dominios. (AyushExel, 2023)

4.9 Scanner advanced ip scanner

Advanced Scanner es un instrumento de escaneo de la red gratuita para Windows que sirve para descubrir dispositivos conectados a una red local y obtener información detallada sobre ellos. Su función principal es proporcionar una lista de direcciones IP y dispositivos activos en la red, junto con detalles como nombres de host, direcciones MAC y estado de la conexión. (Luz, 2021)

Su funcionamiento es relativamente sencillo:

- **Escaneo de IP:** Advanced IP Scanner realiza un escaneo de la red en busca de dispositivos activos. Utiliza protocolos estándar como ICMP, NetBIOS, SNMP y TCP para identificar los dispositivos. (Advanced IP Scanner, 2021)
- **Recopilación de información:** Una vez identificados los dispositivos, el programa recopila información adicional, como direcciones IP, nombres de host, direcciones MAC y proveedores de servicios de red. (Advanced IP Scanner, 2021)
- **Interfaz de usuario:** La información se presenta en una interfaz intuitiva, que da paso a los usuarios a explorar y gestionar los dispositivos detectados.
- **Funciones adicionales:** Además de proporcionar detalles básicos, Advanced IP Scanner también ofrece funciones adicionales, como la capacidad de apagar, reiniciar o enviar mensajes a otros dispositivos en la red. (Advanced IP Scanner, 2021)

Es esencial destacar que Advanced Scanner es una herramienta útil para administradores de red y usuarios que deseen conocer la estructura de su red local. Sin embargo, siempre se debe utilizar con responsabilidad y con el permiso adecuado, ya que el escaneo no autorizado de redes puede infringir políticas de seguridad y privacidad. (Advanced IP Scanner, 2021)

4.9.1 Aplicación realvnc viewer

RealVNC Viewer es una aplicación de software diseñada para permitir el control remoto de un equipo desde otro. Este software utiliza la tecnología de Virtual Network Computing (VNC), que posibilita el acceso y la manipulación de un escritorio de manera remota (Soto, 2020). El funcionamiento básico de RealVNC Viewer es el siguiente:

Servidor VNC: En el equipo que se desea controlar de forma remota, se instala un servidor VNC. Este servidor permite compartir la pantalla y recibir comandos del RealVNC Viewer. (Soto, 2020)

RealVNC Viewer: En el equipo desde el cual se desea controlar de forma remota, se instala el RealVNC Viewer. Este software se utiliza para conectarse al servidor VNC del otro equipo. (Soto, 2020)

Conexión Remota: El usuario inicia RealVNC Viewer e introduce la dirección IP del equipo que desea controlar. Se establece una conexión remota con el servidor VNC del otro equipo. (Soto, 2020)

Control Remoto: Ya determinada la conexión, el usuario puede ver y controlar el escritorio del otro equipo como si estuviera físicamente presente frente a él. Esto incluye el movimiento del mouse, la entrada de teclado y la visualización de la pantalla. (Soto, 2020)

RealVNC Viewer es utilizado comúnmente para soporte técnico remoto, administración de servidores, y en situaciones donde es necesario acceder a un equipo desde una ubicación diferente. La seguridad es importante al utilizar este tipo de software, y se recomienda configurar contraseñas fuertes y, si es posible, utilizar conexiones seguras mediante cifrado. (Soto, 2020)

4.9.2 Tkinter

Tkinter es una biblioteca de Python que se utiliza para la creación de interfaces gráficas de usuario (GUI). Es una de las herramientas más comunes y simples para crear ventanas, botones, menús, cuadros de texto y otros elementos de interfaz gráfica en aplicaciones Python. Tkinter se basa en la biblioteca Tcl/Tk, que proporciona una serie de herramientas para crear usuarios de forma rápida y fácil. (LIB TKINTER, 2023)

4.9.3 Almacenamiento de datos

El almacenamiento de datos (dataset) es un conjunto habitualmente de datos tabulados, cada columna de la tabla corresponde a una variable de la fila que representa los diferentes registros, estos datos se seleccionan y se van almacenando ver la Figura 13 con la intención de realizar investigaciones, análisis y tomar decisiones estratégicas, se puede extraer conocimientos y patrones ocultos, lo que

ayudará a comprender mejor fenómenos e identificar tendencias basadas en evidencias, el almacenamiento de datos suelen ser utilizados en estadística, investigación científica, análisis de datos y machine learning. (DeustoFormación, 2022).

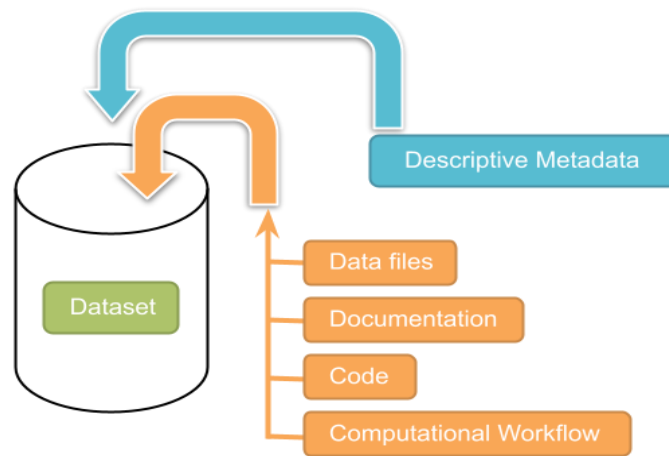


Figura 14 Dataset almacenamiento de datos (Veloso, 2019)

4.9.4 Características del Almacenamiento de datos

Las características de un almacenamiento de datos (dataset) pueden variar ampliamente dependiendo de su naturaleza. Algunas características comunes incluyen la cantidad de datos (tamaño del dataset), el tipo de datos que contiene (numéricos, categóricos, texto, imágenes, etc.), la calidad de los datos (precisión, integridad), la fuente de los datos, y la forma en que están organizados (en tablas, archivos, bases de datos, etc.). Además, pueden contener metadatos que describen los datos, como nombres de columnas, etiquetas, descripciones. (DeustoFormación, 2022)

4.9.5 Cámara ov5647

Esta cámara tiene la capacidad para capturar videos de 1080p e imágenes con enfoque fijo y ajustable se muestra en la Figura 14, tiene un sensor con una resolución de 5 megapíxeles incorpora además dos sensores de leds infrarrojos esto permitirá que tenga una buena visión nocturna, además viene incorporado una fotorresistencia que ayudara a detectar la luz ambiental y la intensidad de la misma, así poder concertar el umbral del infrarrojo automáticamente como se podemos observar en la (UNITElectronics, 2023).



Figura 15 Cámara OV5647 captura videos e imágenes (Cytron.io, 2023)

4.9.6 Sus características principales incluyen:

Es una cámara versátil y ampliamente utilizada en proyectos que requieren captura de imágenes y video de calidad con dispositivos de tamaño reducido.

- **Resolución:** Capacidad para capturar imágenes de 5 megapíxeles (2592x1944 píxeles). **Tamaño de píxel:** Los píxeles tienen un tamaño de 1.4 micrómetros, lo que permite una buena calidad de imagen para su tamaño.

- **Captura de video:** Capacidad para grabar video en diferentes resoluciones, incluyendo Full HD (1080p) a 30 fps y HD (720p) a 60 fps.
- **Interfaces:** Conectividad a través de interfaces como MIPI CSI-2 (Mobile Industry Processor Interfaz, interfaz de procesador de la industria móvil).
- **Aplicaciones:** Utilizada comúnmente en proyectos de robótica, visión artificial, fotografía y aplicaciones de IoT debido a su tamaño compacto y su capacidad para integrarse en sistemas embebidos. (UNITElectronics, 2023)

4.9.7 Raspberry Pi 4

El Raspberry Pi 4 es un pequeño Ordenador portátil que sirve para facilitar la programación, consiste en una placa base se evidencia Figura 15 que soporta distintos componentes electrónicos y funcionan como un procesador ARM, no tiene un circuito de encendido y apagado este microcomputador tiene un agotamiento bajo, pero se debe conectar a corriente por medio de un cargado, esto nos ayuda a realizar proyectos diferentes y es muy utilizado en pequeños prototipos (Calvo, 2022).



Figura 16 Raspberry Pi 4 placa base (Conrad)

4.9.8 Características principales de la raspberry pi 4 son:

- **Procesador:** CPU Broadcom BCM2711 de cuatro núcleos Cortex-A72 a 1.5 GHz, lo que proporciona un incremento de notorio en el rendimiento en comparación con las versiones precedentes.

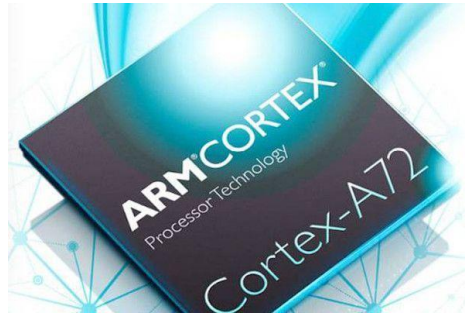


Figura 17 procesador cortex (Muycomputer, 2015)

- **Memoria RAM:** Disponible en variantes de 2GB, 4GB o 8GB de memoria LPDDR4, lo que mejora la capacidad de multitarea y el rendimiento en general.

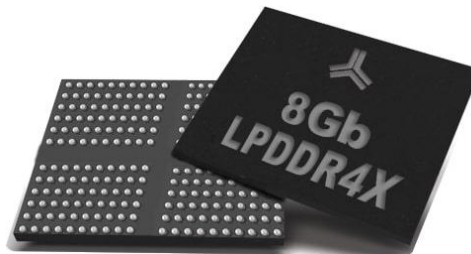


Figura 18 memoria RAM (diarioelectrónico, 2021)

- **Conectividad:** Puertos USB 3.0 y USB 2.0, puertos Gigabit Ethernet, Wi-Fi 802.11ac de doble banda, Bluetooth 5.0, y puertos HDMI para conexión a monitores.

- **Gráficos:** Soporte para resolución de video hasta 4K a través de un puerto HDM



Figura 19 Gráfico 4k (Alamy, 2023)

- **Almacenamiento:** Ranura para tarjetas microSD para el sistema operativo y opciones para conectar unidades de almacenamiento externo.
- **Interfaz:** Varias interfaces GPIO (General Purpose Input/Output) para la conexión de periféricos y sensores.
- **Sistema Operativo:** Compatible con diferentes sistemas operativos, incluyendo Raspbian (basado en Debian), Ubuntu u otros sistemas Linux.

4.9.9 Python

Python es desde instrumentos de programación hasta software de computadoras, desarrollo web y aplicaciones móviles observe la Figura 19, además nos permite la opción de realizar comentarios para tener recordatorios para futuras funciones o también indicar problemas en las líneas de código (Londoño, 2023).



Figura 20 Python instrumento de programación (Alamy, 2021)

4.9.9.1 Características de Python

Python se clasifica como una expresión interpretada, lo que implica ejecutar el código de manera secuencial, abarcando diversas aplicaciones que van desde el desarrollo de la web hasta el análisis de los datos, del aprendizaje automático y de la inteligencia artificial, destacándose por sus características distintivas;

- Un lenguaje de uso sencillo.
- Un lenguaje con tipado dinámico.
- Un lenguaje de alto nivel.
- Un lenguaje basado en la orientación a objetos.

4.9.9.2 Función del router

El dispositivo como el router ofrece conectividad Wi-Fi y suele estar enlazado a un módem. Facilita la transmisión de datos desde el Internet a conexión de otros dispositivos personales como teléfonos, computadoras o tablets. Ver figura 20. La

red en tu hogar está compuesta por los dispositivos conectados a Internet, y el router se encarga de distribuir la información obtenida por el módem a estos dispositivos personales. (AVG, 2022)



Figura 21 Router

V MARCO METODOLÓGICO

5.1 Desarrollo del proyecto

Para la realización del prototipo se va a entrenar un modelo de detección de plagas, mediante la recopilación muestras de imágenes digitales utilizando una cámara que ayudará a captar alteraciones en el cultivo de arroz y poder descubrir a tiempo las posibles plagas que estén afectando al cultivo.

Se utilizará un método experimental en el cual se va a recolectar muestras de imágenes mediante una cámara, que se almacenarán a un almacenamiento con el propósito de registrar diversos datos representativos, con esto se incluirán las imágenes de las plagas que afectan en distintas etapas del cultivo del arroz, se dividirá en conjuntos entrenamiento, entrenamiento, prueba y validación aplicando un lenguaje de código de programación y reconocimiento en Python, además se utilizará Yolov8 junto a OpenCV para el desarrollo de aprendizaje y entrenamiento del modelo con el fin de evaluar segmentos, patrones y características en las imágenes para su rendimiento y obtener registros para la base de datos.

El modelo que se va a entrenar para su función, se adjuntará la cámara OV5647 que capturará las imágenes de tiempo real con visión de computadora y a su vez procesará la programación del modelo que se va a entrenar para ejecutar la detección de las plagas.

5.2 Técnicas de investigación

Para este proyecto se utilizaron las siguientes técnicas:

Técnica experimental: Esta información se logró obtener de muestras recolectadas mediante la observación de nuestro objeto de estudio.

De campo: Mediante visitas de campo se observó y analizó las plagas que afecta a los cultivos.

5.3 Muestra

La selección de las muestras se enfoca en las plagas que afectan directamente a los cultivos especialmente a la de arroz, entre las que participan en este proyecto son:

- La cinta amarilla; muestras utilizadas (92)
- La Mariposa blanca; muestras utilizadas (92)

Diagrama de fases para la elaboración del prototipo

Para realizar este proyecto se seguirá del diagrama de fases que se planteó en la Figura 21, se evidencia el proceso de las etapas en que se llevará a cabo para detectar las plagas.

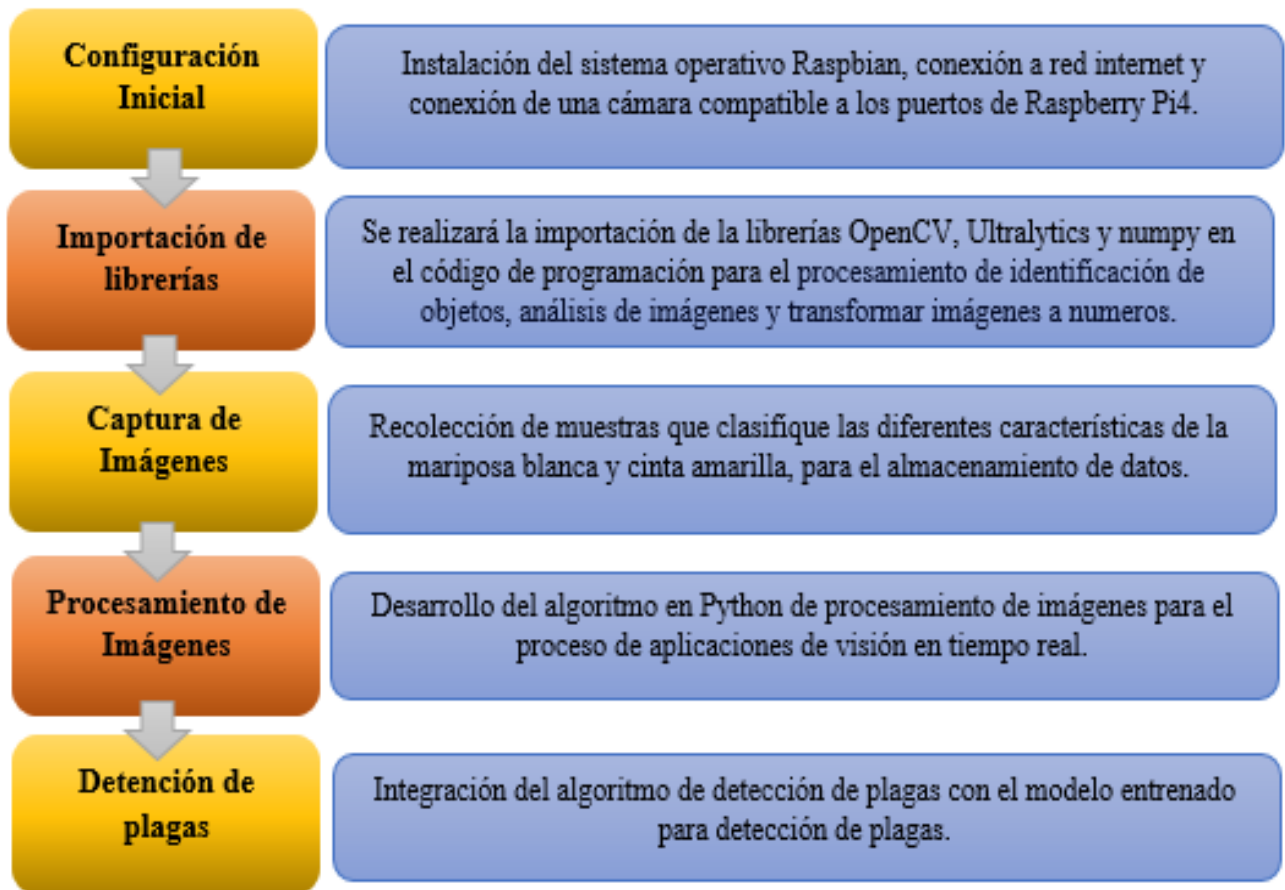


Figura 22 Diagrama de fases para la elaboración del prototipo

Diseño del prototipo

El diseño que se implementará se guiará por el diagrama de fases, ver figura 22, muestra la forma en que se realizará las conexiones del prototipo.

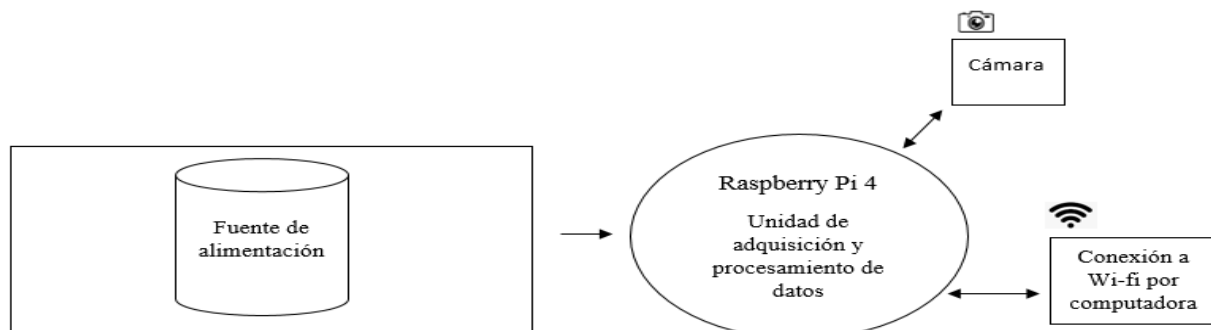


Figura 23 diseño del prototipo

5.4 Desarrollo del prototipo para la detección de plagas en los cultivos de arroz.

Para la elaboración del prototipo, como se observa en la figura 23. Se colocó la raspberry pi 4 en un caset con disipador de calor con el objetivo que la tarjeta no quede expuesta y no se sobrecaliente, además se conectó el Flex de la cámara a la tarjeta para su función para ello ver figura 23, figura 24. se utilizó un router el cual permitirá conectarse remotamente mediante wi-fi que servirá ver el ordenador así permita controlar el monitoreo a través de una PC.

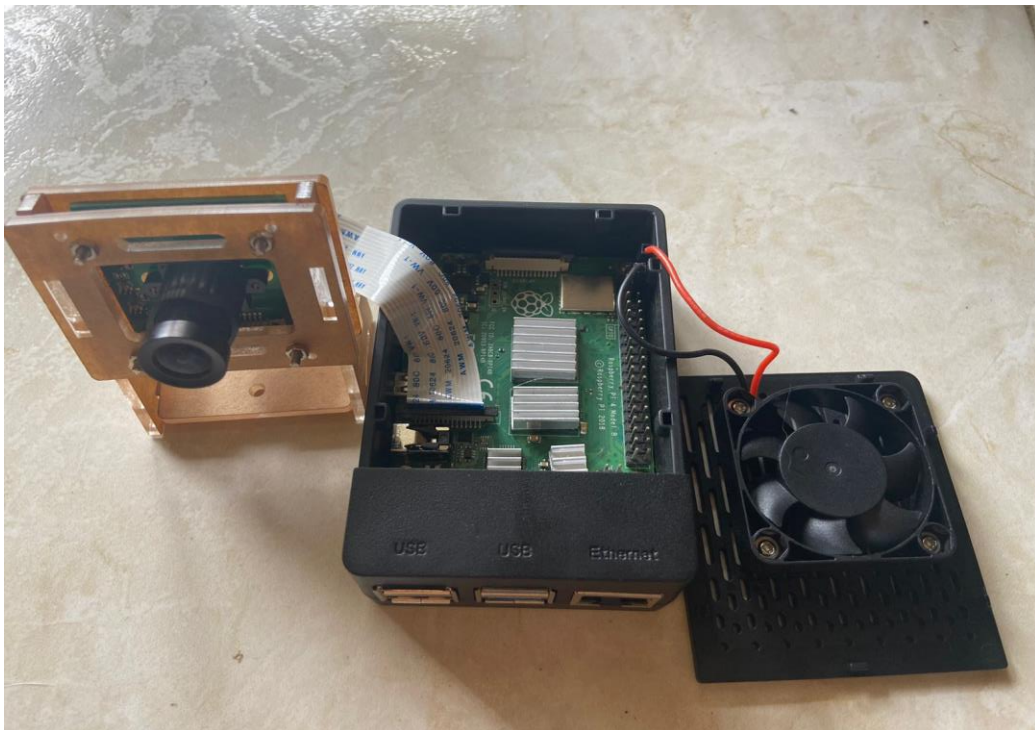


Figura 24 conexión del Flex de la cámara al raspberry pi 4 en caset con disipador de calor



Figura 25 Router para conexión remota hacia la raspberry pi 4

5.4.1 Instalación del sistema operativo

Para el desarrollo del prototipo se requirió instalar Raspbian que es el sistema operativo el cual se encuentra en la página oficial de Raspberry pi, se procedió a descargar el archivo del sistema operativo para utilizar Escher donde se gravó la imagen de Raspbian en una microSD de 64GB.

Después de grabar la imagen se abrió la partición de arranque de la tarjeta microSD en la computadora creando un archivo en el directorio raíz de la partición de arranque, llamado:

wpa_supplicant.conf

Por medio de este comando procedemos a tener lo que la conexión por wifi se procede a editar el archivo wpa_supplicant.conf e ingresa la configuración de tu red Wi-Fi.

```
country=ES
```

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
```

```
update_config=1
```

```
network={
```

```
    ssid="BURGOS_MUÑIZ"
```

```
    psk="47129092"
```

```
}
```

Se guardo este archivo y se procedió a expulsar la microSD. Una vez completado esto se obtuvo acceso a la Raspberry Pi 4 con el Raspbian instalado, configuración inicial realizada y conexión a internet. Ver figura 25.

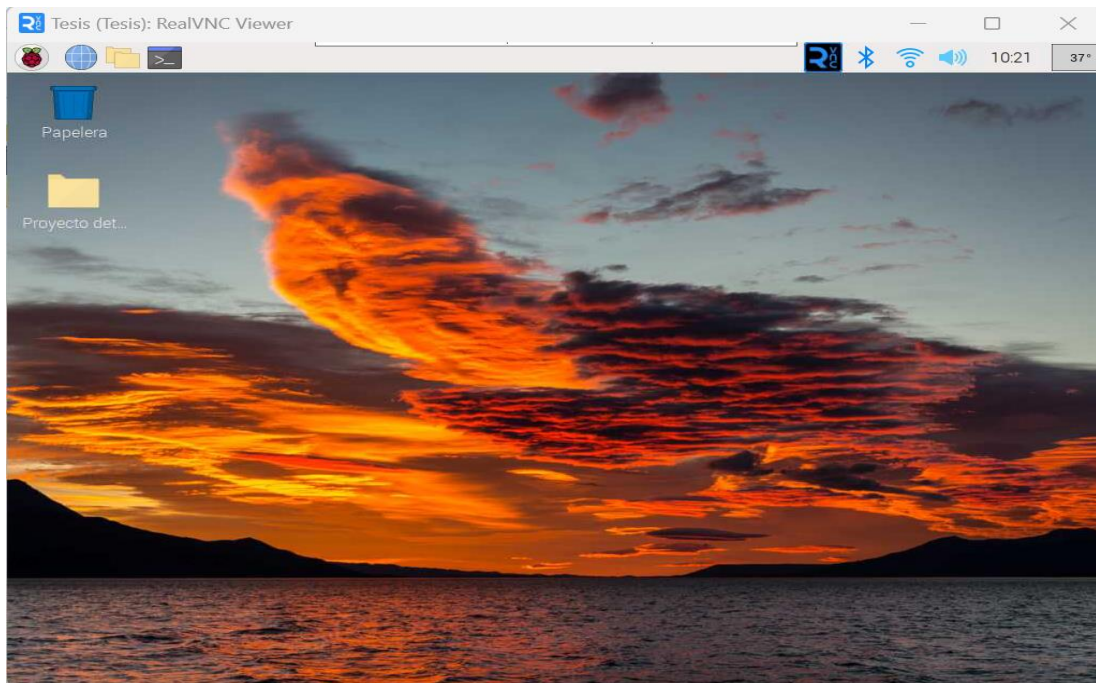


Figura 26 sistema operativo Raspbian instalado

5.4.2 Proceso de conexión por medio del Router

Se realizó una conexión remotamente de la raspberry pi 4 a la PC a través de dos aplicaciones, Advanced IP Scanner y RealVNC Viewer, en donde se procede hacer los siguientes pasos:

- ✓ Se enciende el router, para que la raspberry pi y la PC puedan conectarse a la misma red Wi-fi.
- ✓ Se abre Advanced IP Scanner para scanear los dispositivos que se encuentren conectados.

- ✓ Se debe obtener la dirección IP de la raspberry pi, ver Figura 26.

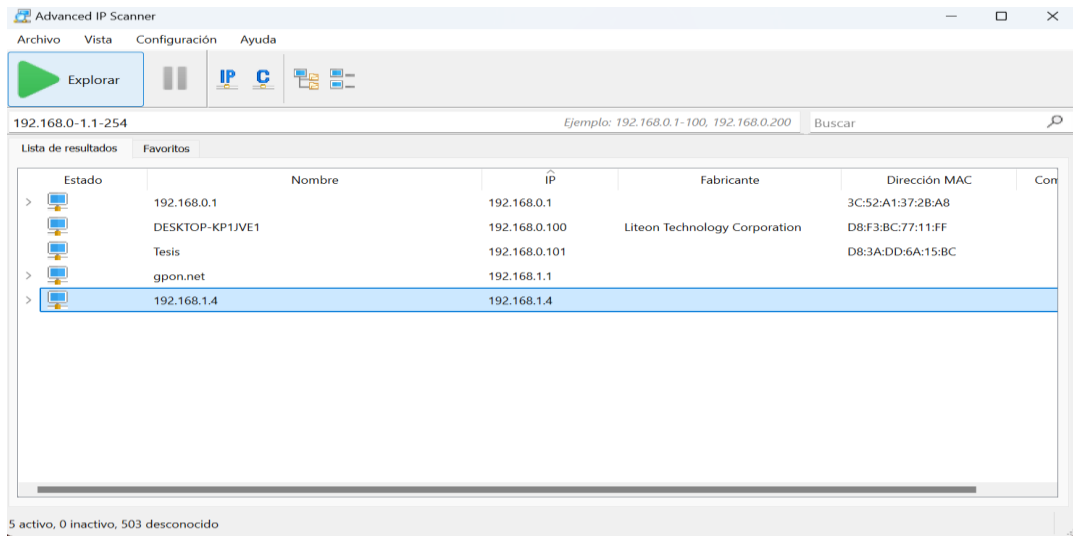


Figura 27 Dirección IP de la Raspberry pi 4

- ✓ Se ejecuta la aplicación RealVNC Viewer y se da clic en nuevo archivo para tener nueva conexión.
- ✓ Se colocan las propiedades del dispositivo que vamos a agregar junto con la dirección IP de la raspberry pi, ver Figura 27.

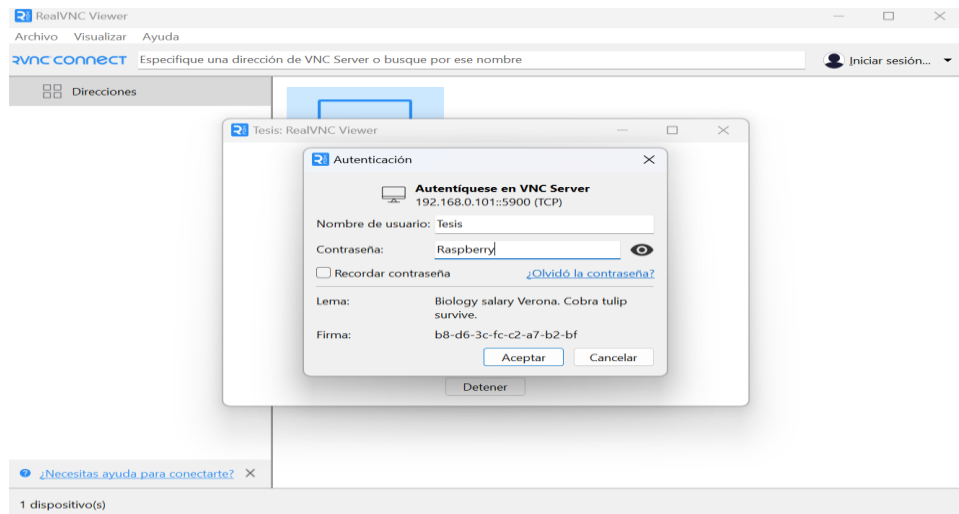


Figura 28 propiedades del dispositivo raspberry pi

Se abre la pestaña de autenticación de usuario y contraseña de usuario para acceder al dispositivo de forma remota, ver Figura 28, figura 29.

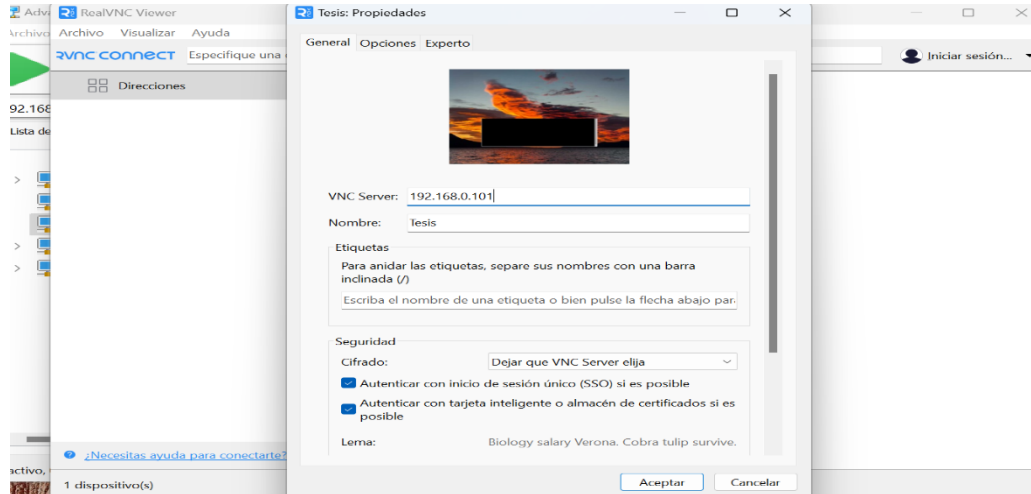


Figura 29 pestaña de autenticación de usuario y contraseña

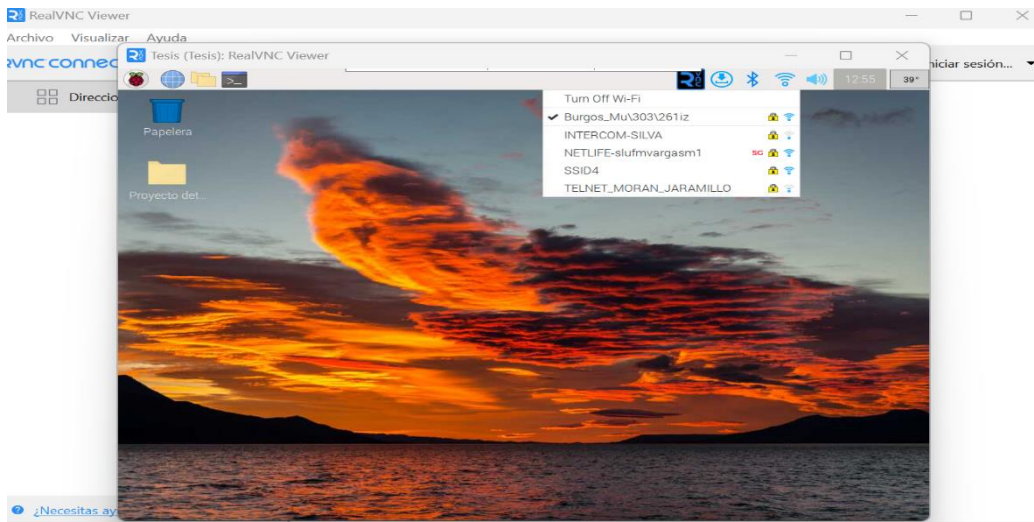


Figura 30 conexión remota a red Wi-fi

5.4.3 Adquisición de imágenes

En la adquisición de imágenes se realizó una visita al campo para recolectar diversas muestras de las plagas del cultivo de arroz a través de una cámara.

Entre dichas muestras se recolecto principalmente muestras de las plagas con las que vamos a trabajar que son, cinta amarilla y mariposa blanca, Además se optó por seleccionar unas muestras del cultivo sano para clasificarlas con las demás. Y así tener muestras tanto de escenarios afectas como escenarios sanos, ver figura 30.

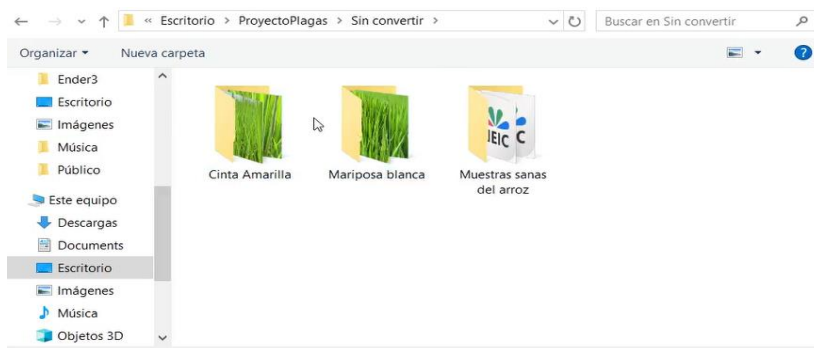


Figura 31 recolección de muestras de las plagas

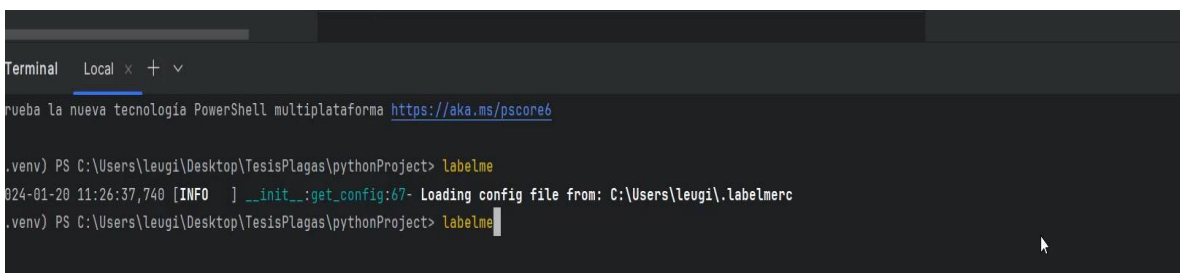
VI RESULTADOS

Para la elaboración del prototipo de visión artificial se utilizó la biblioteca de Open CV, Yolo. Un algoritmo que se desarrolló en Python para la detección de objetos en imágenes y videos en este caso las plagas, adicional se utilizó Pycharm que ayuda al desarrollo proyectos escritos en Python esto tiene como resultado lo siguiente:

- Creación del almacenamiento de datos con las muestras obtenidas de las plagas más comunes; cinta amarilla y mariposa blanca.
- Entrenamiento del modelo en base al almacenamiento de datos creados.
- Implementación y programación para el modelo de detención de plagas en la Raspberry pi 4.

6.1 Creación del almacenamiento de datos con las muestras obtenidas de las plagas más comunes; cinta amarilla y mariposa blanca.

Para la creación del almacenamiento de datos, se empleó lo que es una herramienta llamada labelme la cual se ejecuta a través del siguiente comando: ver figura 31



```
Terminal Local x + v
prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject> labelme
2024-01-20 11:26:37,740 [INFO ] __init__:get_config:67- Loading config file from: C:\Users\leugi\labelmerc
.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject> labelme
```

Figura 32 Comando para ejecutar labelme

Una vez ejecutado el programa se abrirá la siguiente ventana se muestra en la figura 32, 33, 34, 35 que permite seleccionar la carpeta de las imágenes que luego nos permitió hacer los etiquetados de cada una de las muestras obtenidas de las plagas.

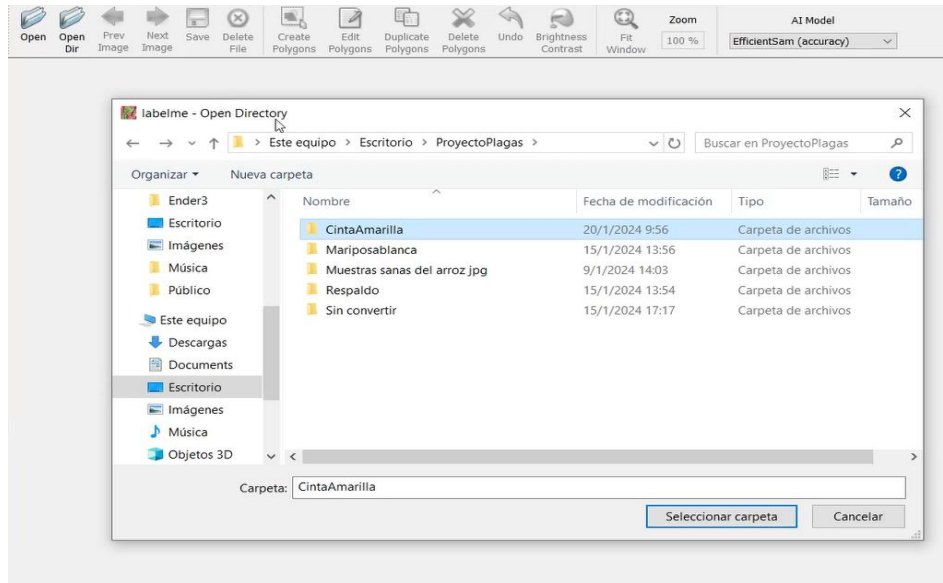


Figura 33 Carpeta de imágenes seleccionadas

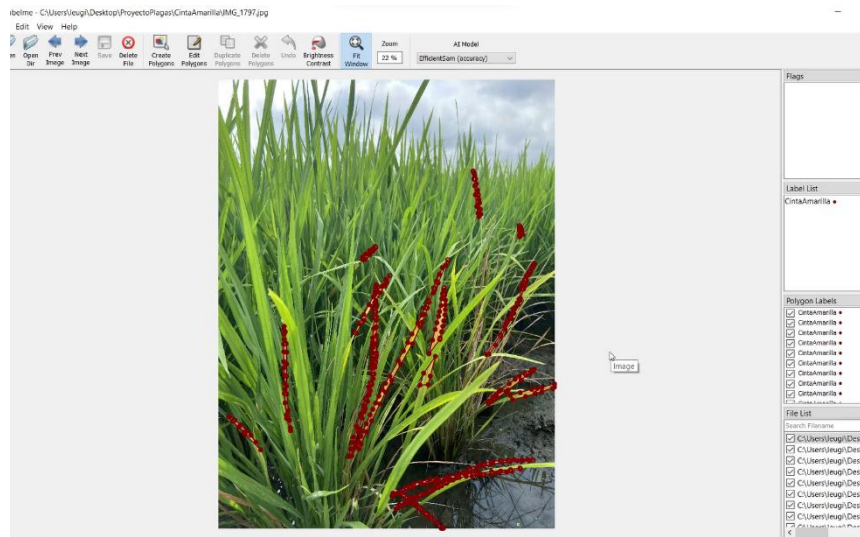


Figura 34 muestra afectada cinta amarilla

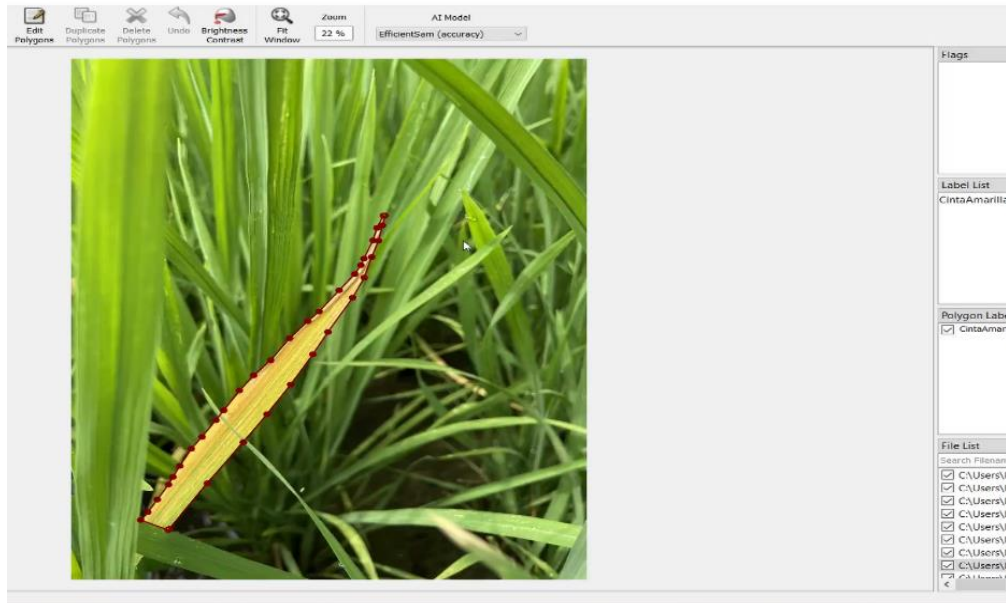


Figura 35 etiquetado de la muestra afectada

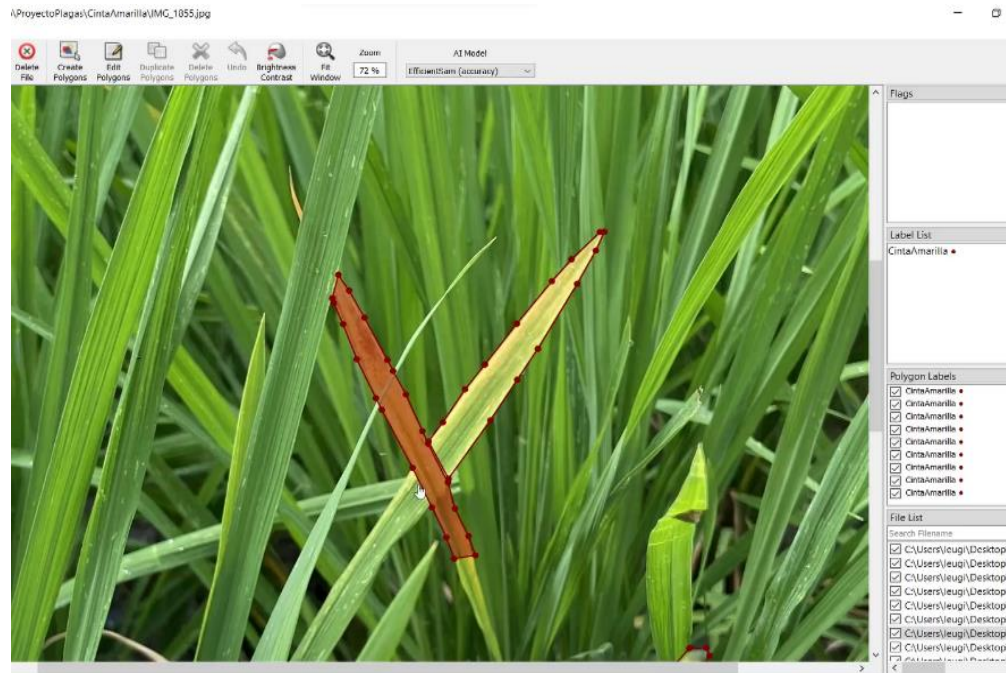


Figura 36 etiquetado de la muestra

En la figura. Se observa el respectivo etiquetado en cada una de las muestras, tanto de cinta amarilla como la de mariposa blanca. Ver figura 36, 37

Escritorio > ProyectoPlagas > CintaAmarilla >

Nombre	Fecha de modificación	Tipo	Tamaño
IMG_1801	19/1/2024 18:49	Archivo de origen ...	2,274 KB
IMG_1802	9/1/2024 13:52	Archivo JPG	1,319 KB
IMG_1802	19/1/2024 20:01	Archivo de origen ...	2,408 KB
IMG_1803	9/1/2024 13:52	Archivo JPG	1,138 KB
IMG_1803	20/1/2024 3:46	Archivo de origen ...	2,070 KB
IMG_1804	9/1/2024 13:50	Archivo JPG	1,441 KB
IMG_1804	19/1/2024 20:56	Archivo de origen ...	2,619 KB
IMG_1805		Archivo JPG	1,389 KB
IMG_1805		Archivo de origen ...	2,537 KB
IMG_1806	9/1/2024 13:50	Archivo JPG	839 KB
IMG_1806	19/1/2024 22:46	Archivo de origen ...	1,500 KB
IMG_1812	9/1/2024 13:50	Archivo JPG	1,061 KB
IMG_1812	19/1/2024 22:51	Archivo de origen ...	1,923 KB
IMG_1813	9/1/2024 13:51	Archivo JPG	1,075 KB
IMG_1813	20/1/2024 3:58	Archivo de origen ...	1,948 KB
IMG_1820	9/1/2024 13:51	Archivo JPG	632 KB
IMG_1820	19/1/2024 23:00	Archivo de origen ...	1,100 KB
IMG_1821	9/1/2024 13:51	Archivo JPG	606 KB
IMG_1821	20/1/2024 3:59	Archivo de origen ...	1,046 KB
IMG_1822	9/1/2024 13:51	Archivo JPG	1,089 KB
IMG_1822	19/1/2024 23:07	Archivo de origen ...	1,956 KB
IMG_1823	9/1/2024 13:51	Archivo JPG	818 KB
IMG_1823	19/1/2024 23:19	Archivo de origen ...	1,455 KB
IMG_1824	9/1/2024 13:51	Archivo JPG	610 KB
IMG_1824	19/1/2024 23:24	Archivo de origen ...	1,062 KB
IMG_1853	9/1/2024 13:51	Archivo JPG	948 KB
IMG_1853	20/1/2024 0:45	Archivo de origen ...	1,706 KB
IMG_1854	9/1/2024 13:51	Archivo JPG	978 KB
IMG_1854	20/1/2024 0:57	Archivo de origen ...	1,765 KB
IMG_1855	9/1/2024 13:51	Archivo JPG	901 KB
IMG_1855	20/1/2024 1:11	Archivo de origen ...	1,620 KB
IMG_1856	9/1/2024 13:51	Archivo JPG	864 KB
IMG_1856	20/1/2024 1:20	Archivo de origen ...	1,545 KB
IMG_1857	9/1/2024 13:51	Archivo JPG	493 KB

Tipos de archivos de origen JSON:
 Tipo: Archivo de origen JSON
 Tamaño: 2.55 MB
 Fecha de modificación: 19/1/2024 20:56

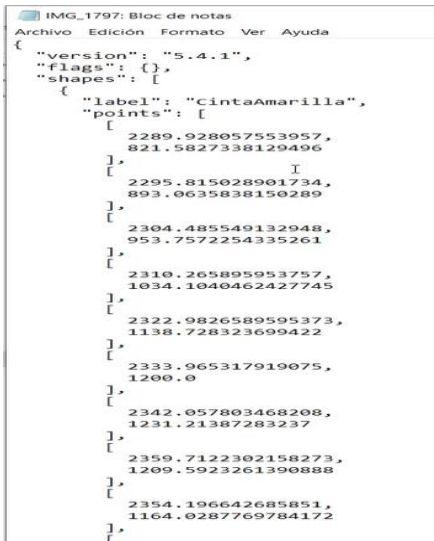
Figura 37 etiquetado de cada imagen cinta amarilla

Escritorio > ProyectoPlagas > Mariposablanca >

Nombre	Fecha de modificación	Tipo	Tamaño
IMG_1919	9/1/2024 13:56	Archivo JPG	899 KB
IMG_1919	15/1/2024 2:08	Archivo de origen ...	1,621 KB
IMG_1920	9/1/2024 13:56	Archivo JPG	834 KB
IMG_1920	15/1/2024 2:16	Archivo de origen ...	1,489 KB
IMG_1923	9/1/2024 13:56	Archivo JPG	868 KB
IMG_1923	15/1/2024 2:21	Archivo de origen ...	1,556 KB
IMG_1924	9/1/2024 13:56	Archivo JPG	1,027 KB
IMG_1924	15/1/2024 2:27	Archivo de origen ...	1,855 KB
IMG_1925	9/1/2024 13:56	Archivo JPG	1,076 KB
IMG_1925	15/1/2024 2:32	Archivo de origen ...	1,949 KB
IMG_1929	9/1/2024 13:56	Archivo JPG	1,029 KB
IMG_1929	15/1/2024 2:41	Archivo de origen ...	1,858 KB
IMG_1933	9/1/2024 13:56	Archivo JPG	1,284 KB
IMG_1933	15/1/2024 2:53	Archivo de origen ...	2,337 KB
IMG_1935	9/1/2024 13:56	Archivo JPG	760 KB
IMG_1935	15/1/2024 3:13	Archivo de origen ...	1,364 KB
IMG_1937	9/1/2024 13:56	Archivo JPG	948 KB
IMG_1937	15/1/2024 3:20	Archivo de origen ...	1,698 KB
IMG_1938	9/1/2024 13:56	Archivo JPG	839 KB
IMG_1938	15/1/2024 3:36	Archivo de origen ...	1,505 KB
IMG_1942	9/1/2024 13:56	Archivo JPG	1,145 KB
IMG_1942	15/1/2024 3:44	Archivo de origen ...	2,081 KB
IMG_1943	9/1/2024 13:56	Archivo JPG	1,415 KB
IMG_1943	15/1/2024 4:02	Archivo de origen ...	2,582 KB
IMG_1944	9/1/2024 13:56	Archivo JPG	1,423 KB
IMG_1944	15/1/2024 4:14	Archivo de origen ...	2,581 KB
IMG_1945	9/1/2024 13:56	Archivo JPG	1,272 KB
IMG_1945	15/1/2024 4:23	Archivo de origen ...	2,306 KB
IMG_1947	9/1/2024 13:56	Archivo JPG	1,215 KB
IMG_1947	15/1/2024 5:29	Archivo de origen ...	2,224 KB
IMG_1948	9/1/2024 13:56	Archivo JPG	1,030 KB
IMG_1948	15/1/2024 5:39	Archivo de origen ...	1,871 KB
IMG_1949	9/1/2024 13:56	Archivo JPG	910 KB
IMG_1949	15/1/2024 5:51	Archivo de origen ...	1,644 KB

Figura 38 etiquetado de cada imagen mariposa blanca

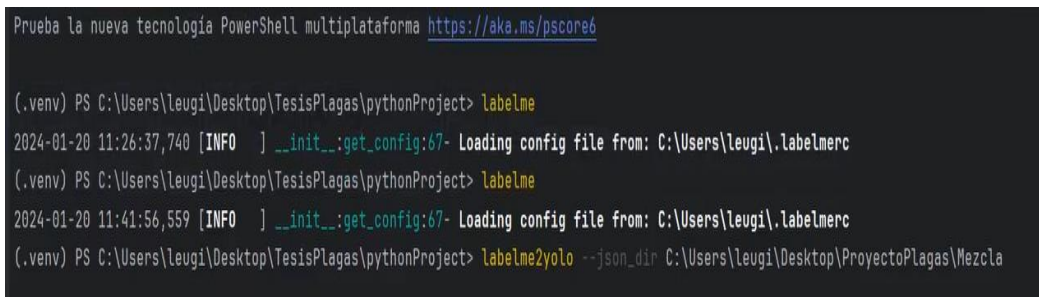
Ejemplo de unos de los etiquetados que se abrió, se encuentra cada uno de los puntos rojos que se muestran en la imagen esto sirvió como referencia para hacer el entrenamiento. Ver figura 38.



```
IMG_1797: Bloc de notas
Archivo Edición Formato Ver Ayuda
{
  "version": "5.4.1",
  "flags": {},
  "shapes": [
    {
      "label": "CintaAmarilla",
      "points": [
        [
          2289.928057553957,
          821.5827338129496
        ],
        [
          2295.815028901734,
          893.0635838150289
        ],
        [
          2304.485549132948,
          953.7572254335261
        ],
        [
          2310.265895953757,
          1034.1040462427745
        ],
        [
          2322.9826589595373,
          1138.728323699422
        ],
        [
          2333.965317919075,
          1200.0
        ],
        [
          2342.057803468208,
          1231.21387283237
        ],
        [
          2359.7122302158273,
          1209.5923261390888
        ],
        [
          2354.196642685851,
          1164.0287769784172
        ],
        [
          ]
        ]
      }
    ]
  }
}
```

Figura 39 Referencia de los puntos de las imágenes

Se creo una carpeta con las muestras de las imágenes de la cinta amarilla y mariposa blanca a la cual se le dominó mezcla con el objetivo de convertir el formato labelme a Yolo, ver figura 39.



```
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

(.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject> LabelMe
2024-01-20 11:26:37,740 [INFO ] _init_:get_config:67- Loading config file from: C:\Users\leugi\Labelmerc

(.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject> LabelMe
2024-01-20 11:41:56,559 [INFO ] _init_:get_config:67- Loading config file from: C:\Users\leugi\Labelmerc

(.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject> LabelMe2yolo --json_dir C:\Users\leugi\Desktop\ProyectoPlagas\Mezcla
```

Figura 40 Comando de conversión de labelme a Yolo

Este proceso de conversión tiene un tiempo de duración de aproximadamente 10 minutos y una vez terminada la conversión se obtuvo el almacenamiento de datos. Ver figura 40, 41

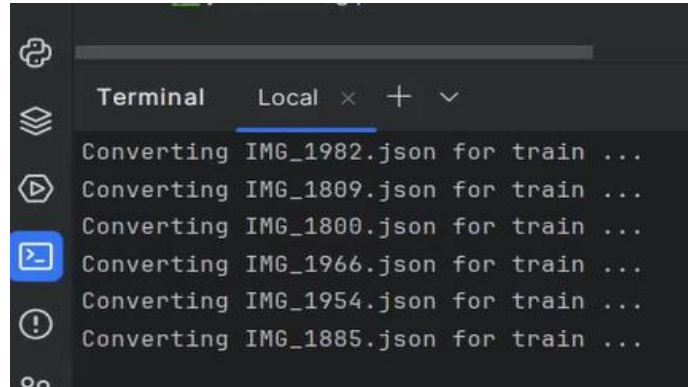


Figura 41 Conversión de las muestras

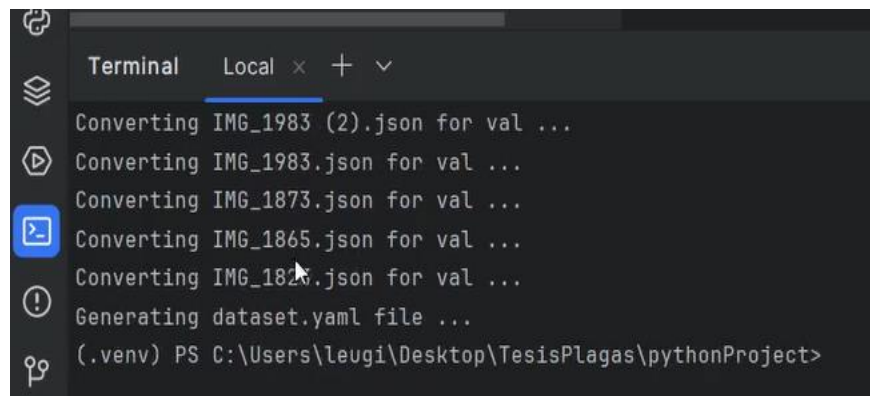


Figura 42 conversión finalizada

Terminado el proceso de conversión automáticamente muestra la carpeta de mezcla el archivo de conversión YoloDataset obteniendo nuestra base de datos para el modelo. Ver figura 42.

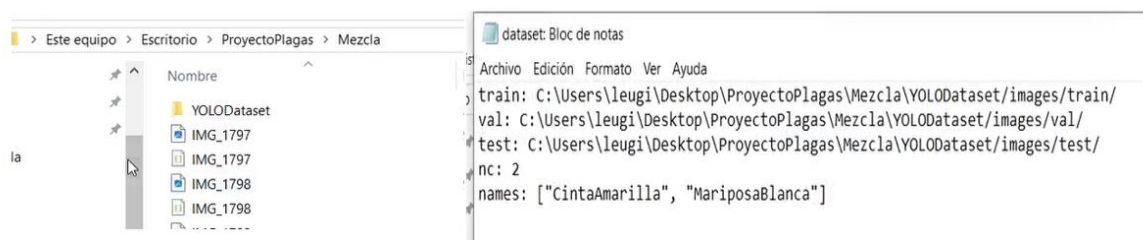
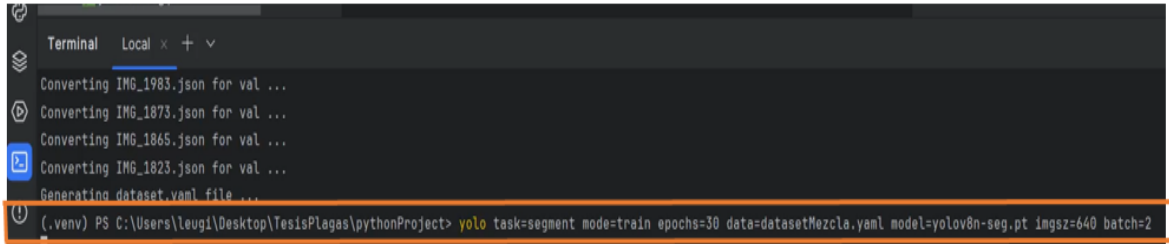


Figura 43 Almacenamiento de datos

6.2 Entrenamiento del modelo en base al almacenamiento de datos creados.

Para realizar el entrenamiento del modelo se realizó con un comando. Ver figura 43

A screenshot of a terminal window with a dark background. The window title is "Terminal Local x + v". The terminal shows several lines of output: "Converting IMG_1983.json for val ...", "Converting IMG_1873.json for val ...", "Converting IMG_1865.json for val ...", "Converting IMG_1823.json for val ...", and "Generating dataset.yaml file ...". The last line of output is highlighted with a red rectangular box. Below the highlighted output, the terminal prompt shows the command: "(.venv) PS C:\Users\teugi\Desktop\TesisPlagas\pythonProject> yolo task=segment mode=train epochs=30 data=datasetMezcla.yaml model=yolov8n-seg.pt imgsz=640 batch=2".

```
Terminal Local x + v
Converting IMG_1983.json for val ...
Converting IMG_1873.json for val ...
Converting IMG_1865.json for val ...
Converting IMG_1823.json for val ...
Generating dataset.yaml file ...
(.venv) PS C:\Users\teugi\Desktop\TesisPlagas\pythonProject> yolo task=segment mode=train epochs=30 data=datasetMezcla.yaml model=yolov8n-seg.pt imgsz=640 batch=2
```

Figura 44 Comando para entrenamiento del modelo

Este comando con el que se realizó el modelo ayudó a que las escenas sean más precisas con el propósito de que descifre cual es el objeto en sí, ya que el ambiente en el que se trabajó es el campo por lo que hay diversos factores que afectan al momento de detectar su objetivo. Este modelo se realizó en 30 épocas neuronales que cada una de ellas mandan diferentes respuestas correctas en cada época realizada.

El modelo se realizó en nano ya que es el archivo más ligero, que dura aproximadamente de 15 a 20 min en entrenar el modelo, dependiendo de la cantidad de imágenes. Ver figura 44.

Al terminar este proceso envía la información a una carpeta llamada Train13/Runs

```

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
2/30 0.753G 1.917 2.028 3.357 1.41 1 640: 100% | 60/60 [00:24<00:00, 2.41it/s]
Class Images Instances Box(P R mAP50 mAP50-95) Mask(P R mAP50 mAP50-95): 100% | 7/7 [00:02<00:00,
all 27 69 0.48 0.388 0.361 0.138 0.503 0.404 0.389 0.158

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
5/30 0.623G 1.978 1.949 2.981 1.41 4 640: 100% | 60/60 [00:33<00:00, 1.77it/s]
Class Images Instances Box(P R mAP50 mAP50-95) Mask(P R mAP50 mAP50-95): 100% | 7/7 [00:02<00:00,
all 27 69 0.465 0.538 0.521 0.233 0.707 0.448 0.536 0.289

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
8/30 0.696 2.042 1.858 2.997 1.435 1 640: 100% | 60/60 [00:38<00:00, 1.56it/s]
Class Images Instances Box(P R mAP50 mAP50-95) Mask(P R mAP50 mAP50-95): 100% | 7/7 [00:02<00:00,
all 27 69 0.894 0.469 0.544 0.228 0.87 0.478 0.59 0.321

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
9/30 0.591G 2.127 1.973 2.966 1.548 15 640: 50% | 30/60 [00:19<00:18, 1.65it/s]

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
13/30 0.803G 1.825 1.532 2.523 1.425 1 640: 100% | 60/60 [00:42<00:00, 1.42it/s]
Class Images Instances Box(P R mAP50 mAP50-95) Mask(P R mAP50 mAP50-95): 100% | 7/7 [00:03<00:00,
all 27 69 0.653 0.539 0.556 0.267 0.68 0.534 0.567 0.3

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
14/30 0.763G 1.947 1.58 2.416 1.452 17 640: 43% | 26/60 [00:18<00:21, 1.56it/s]

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
24/30 0.638G 1.655 1.328 2.28 1.341 1 640: 100% | 60/60 [00:39<00:00, 1.52it/s]
Class Images Instances Box(P R mAP50 mAP50-95) Mask(P R mAP50 mAP50-95): 100% | 7/7 [00:02<00:00,
all 27 69 0.574 0.591 0.548 0.265 0.547 0.6 0.54 0.311

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
25/30 0.667G 1.524 1.352 2.188 1.424 4 640: 8% | 5/60 [00:02<00:33, 1.62it/s]
all 27 69 0.746 0.581 0.628 0.32 0.749 0.581 0.643 0.385
CintaAmarilla 27 41 0.695 0.341 0.433 0.219 0.7 0.341 0.464 0.256
MariposaBlanca 27 28 0.797 0.821 0.823 0.421 0.798 0.821 0.823 0.514

Speed: 1.7ms preprocess, 19.3ms inference, 0.1ms loss, 26.0ms postprocess per image
Results saved to runs\segment\train13
Learn more at https://docs.ultralytics.com/modes/train
(.venv) PS C:\Users\leugi\Desktop\TesisPlagas\pythonProject>

```

Figura 45 Épocas de entrenamiento

Se realizaron algunos modelos de entrenamiento ver figura, se elijo el ultimo ya que fue donde se realizó una mezcla de las diferentes muestras recolectadas tanto de cinta amarilla como mariposa blanca. Ver figura 45

> Escritorio > TesisPlagas > pythonProject > runs > segment >

Nombre	Fecha de modificación	Tipo	Tamaño
train	15/1/2024 14:30	Carpeta de archivos	
train2	15/1/2024 14:42	Carpeta de archivos	
train3	15/1/2024 15:22	Carpeta de archivos	
train4	15/1/2024 15:46	Carpeta de archivos	
train5	15/1/2024 16:48	Carpeta de archivos	
train6	15/1/2024 16:49	Carpeta de archivos	
train7	15/1/2024 16:50	Carpeta de archivos	
train8	15/1/2024 17:06	Carpeta de archivos	
train9	17/1/2024 5:52	Carpeta de archivos	
train10	20/1/2024 10:41	Carpeta de archivos	
train11	20/1/2024 12:03	Carpeta de archivos	
train12	20/1/2024 12:06	Carpeta de archivos	
train13	20/1/2024 12:35	Carpeta de archivos	

Figura 46 Entrenamientos realizados

4.3.1 Resultados del entrenamiento del modelo

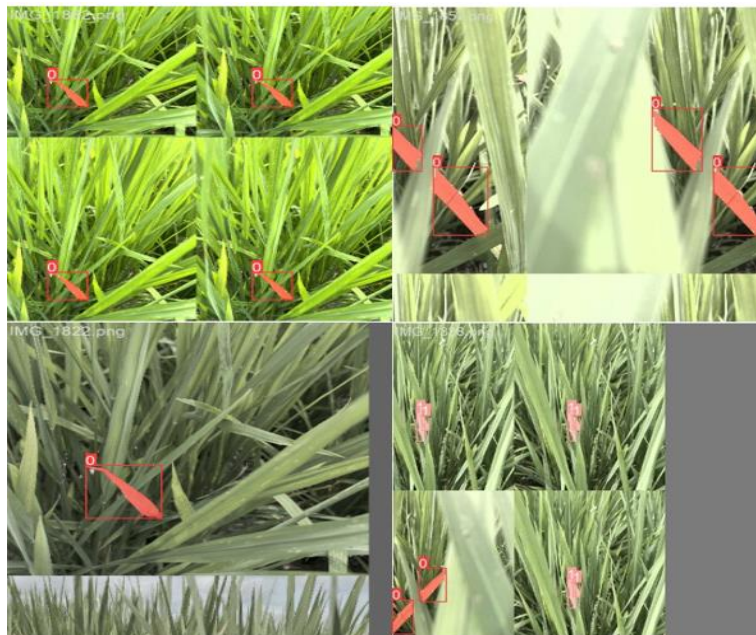


Figura 47 Resultado del entrenamiento del modelo

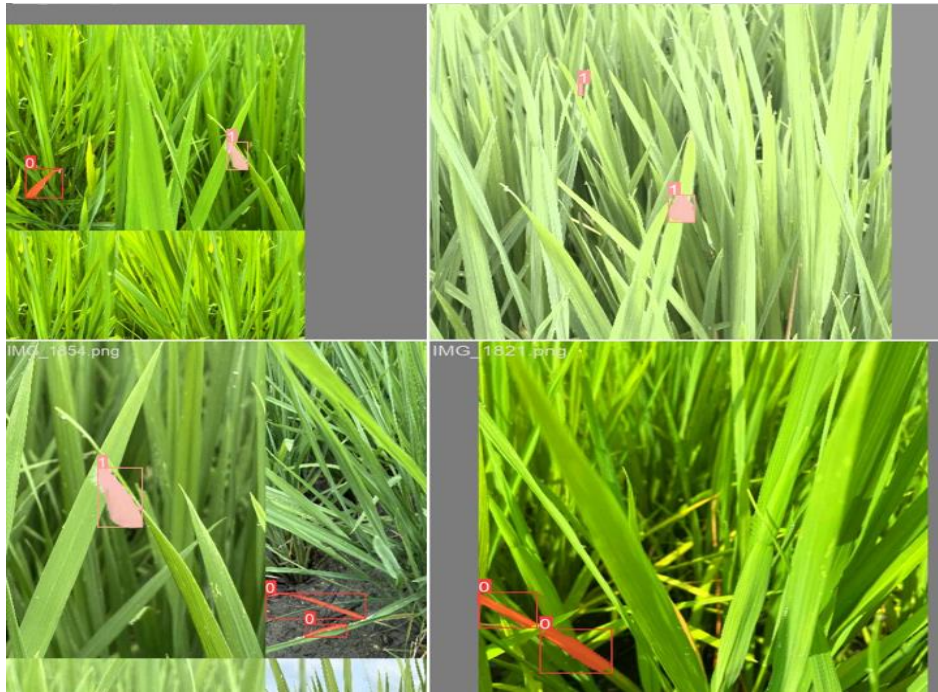


Figura 48 Resultado de detección de la mariposa blanca



Figura 49 detección de mariposa blanca



Figura 50 entrenamiento del modelo cinta amarilla y mariposa blanca



Figura 51 Entrenamiento del modelo mariposa blanca

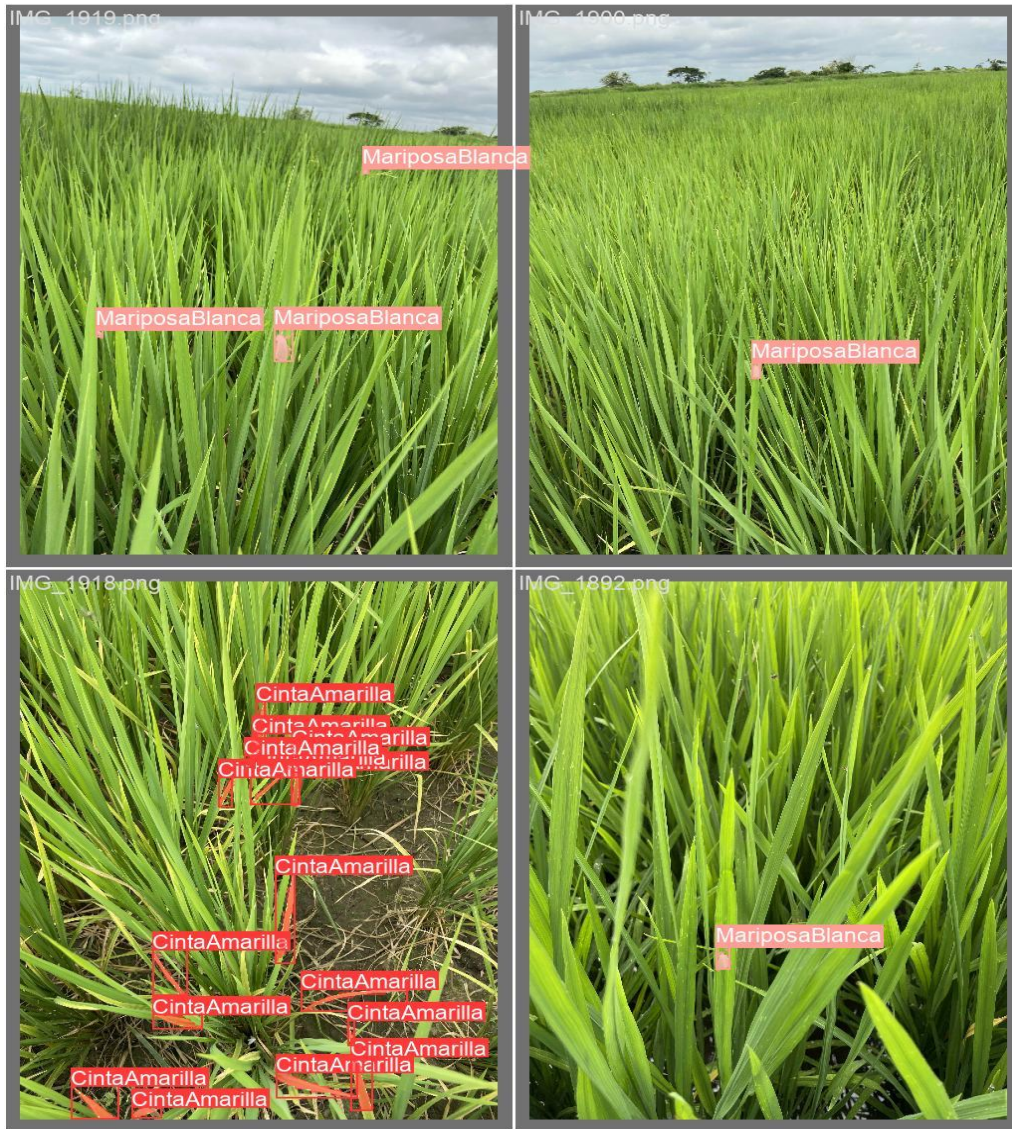
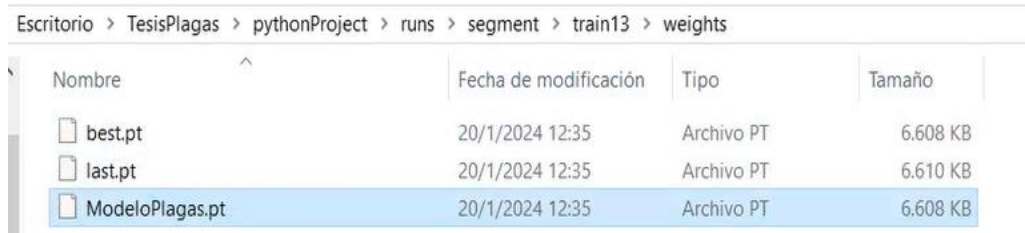


Figura 52 Resultado del entrenamiento del modelo

6.3 Implementación y programación para el modelo de detención de plagas en la tarjeta raspberry pi 4.

Para llevar el modelo entrenado a la raspberry pi 4, se selecciona el archivo ModeloPlagas.pt en la carpeta train13/ weights ver figura 52



Nombre	Fecha de modificación	Tipo	Tamaño
best.pt	20/1/2024 12:35	Archivo PT	6.608 KB
last.pt	20/1/2024 12:35	Archivo PT	6.610 KB
ModeloPlagas.pt	20/1/2024 12:35	Archivo PT	6.608 KB

Figura 53 archivo ModeloPlagas

Se pasa a transferir el modelo del archivo hacia el escritorio del ordenador de la raspberry pi, ver figura 53.

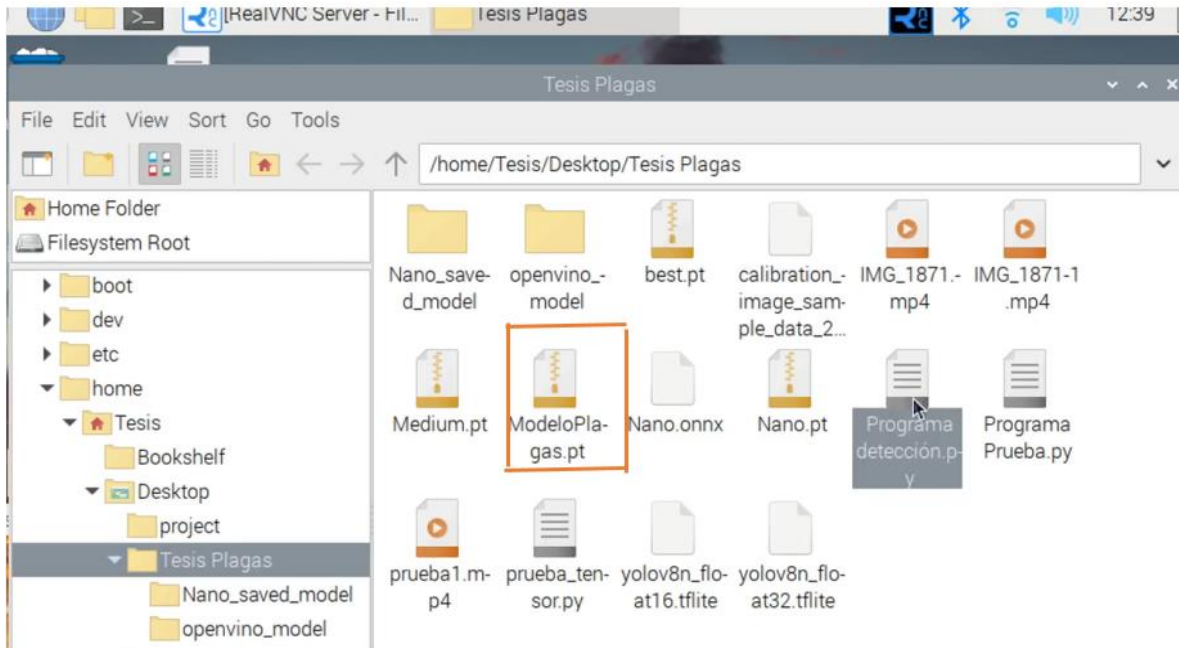


Figura 54 modelo instalado en la raspberry pi 4

Una vez realizada la implementación del modelo, se procedió hacer la programación para la detección de plagas donde se realizarán los siguientes pasos:

- Importación de las librerías con las que vamos a trabajar que son Yolo, Numpy, tkinter, pandas y OpenCV. Ver figura 54.

```
users / tedgi / Desktop / Programa detección GUIz.py / ...
# Importamos las librerías
from ultralytics import YOLO
from tkinter import *
from tkinter import filedialog
from PIL import Image, ImageTk
import imutils
import cv2
import numpy as np
import pandas as pd
```

Figura 55 Código Importación de librerías

- Se realizó la configuración de la interfaz agregándole un título, el logo de la universidad y los botones de funciones, cada una de estas con sus respectivos parámetros para su ubicación. Ver figura 55.

```
titulo = '''UNIVERSIDAD POLITECNICA SALESIANA
"DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA
DETECTAR PLAGAS EN LOS CULTIVOS DE ARROZ UTILIZANDO
UNA RASPBERRY PI Y OPENCV"
AUTOR: Yostin Jahir Burgos Muñiz
TUTOR: Phd. Victor Manuel Huilcapi Subia'''

# Ventana Principal
# Pantalla
pantalla = Tk()
pantalla.title("Programa de Titulacion")
pantalla.geometry("840x700") # Asignamos la dimension de la ventana

# Interfaz
Titulo = Label(pantalla, text=titulo)
Titulo.place(x = 250, y = 5)

# Logo
#Logo = PhotoImage(resize_logo)
log = ImageTk.PhotoImage(resize_logo)
LogoL = Label(pantalla, image = log, text = "Logo")
LogoL.place(x = 180, y = 35, relwidth = 0.08, relheight = 0.08)

botonIniciar = Button(pantalla, text="Iniciar", width = 10, command=inicio)
botonIniciar.place(x=50, y=30)

botonFinalizar = Button(pantalla, text="Finalizar", width = 10, command=fin)
botonFinalizar.place(x=50, y=80)

# Definir nuestras imagenes
on = PhotoImage(file = "ru")
off = PhotoImage(file = "ri")
```

Figura 56 Código de Interfaz

- Se declara la función iniciar dándonos esta la opción de hacer el llamado a la cámara web o seleccionar un video para realizar la detección en este. Ver figura 56

```

# Funcion iniciar
def iniciar():
    global cap
    # Elegimos fuente de video
    if encendidoCam:
        cap = cv2.VideoCapture(0)
    else:
        cap (variable) encendidoCam: bool
    if not encendidoCam and not len(path_file) > 0:
        textoFile.config(text = "Escoja archivo Primero")
    elif not encendidoCam and len(path_file) > 0:
        visualizar()
    elif encendidoCam:
        visualizar()

```

Figura 57 Código función iniciar

- Se selecciona la ruta para hacer el llamado a nuestro modelo entrenado de detección de plagas. Ver figura 57

```

ruta = '/home/Tesis/Desktop/Proyecto detector de Plagas/'
model = YOLO(ruta + 'ModeloPlagas.pt')
# Read the Image
logo = Image.open("/home/Tesis/Desktop/Proyecto detector de Plagas/Logo.png")

```

Figura 58 código de selección del modelo entrenado

- Se realiza la función visualizar, nos permite mostrar los resultados y anotaciones de la imagen o escenario que está analizando el modelo entrenado. Ver figura 58

```

# Funcion Visualizar
def visualizar():
    global cap, pantalla, frame, encendido, count, resultados, Alto1, Ancho1, Alto2, Ancho2
    # Leemos la videocaptura
    if cap is not None:
        ret, frame = cap.read()

        # Si es correcta
        if ret == True:
            (function) resize: Any
            frame = imutils.resize(frame, width=640)

            Alto1, Ancho1 = frame.shape[:2]
            print(Ancho1 ,Alto1)

        if encendido:

            resultados = model.predict(frame, imsz = 640, conf = 0.10)

            boxes = resultados[0].boxes.xyxy.tolist()
            class_ids = resultados[0].boxes.cls.tolist()
            class_names = resultados[0].names
            confidences = resultados[0].boxes.conf.tolist()

            print(boxes)
            print(class_ids)
            print(class_names)
            print(confidences)
            # Mostramos resultados
            anotaciones = resultados[0].plot()
            anotaciones = cv2.cvtColor(anotaciones, cv2.COLOR_BGR2RGB)

```

Figura 59 Función visualizar

- Para desactivar y salir de la interfaz se utiliza el siguiente código ver figura 59

```

# Funcion finalizar
def finalizar():
    cap.release()
    pantalla.destroy
    cv2.destroyAllWindows()
    print("FIN")

```

Figura 60 código para desactivar la interfaz

6.4 Prueba final del prototipo funcionando

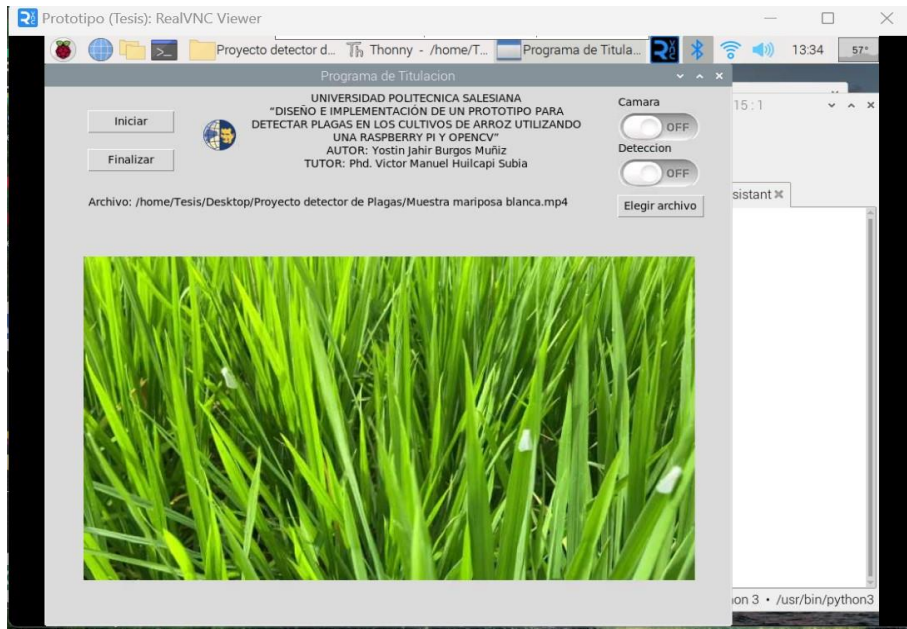


Figura 61 detección de plaga mariposa blanca

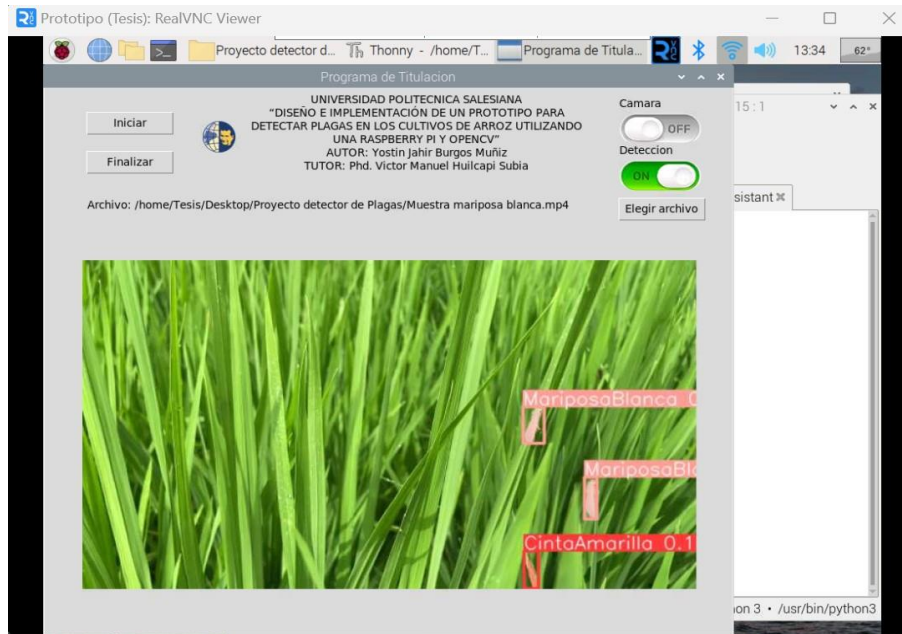


Figura 62 Captura de detección de plaga mariposa blanca

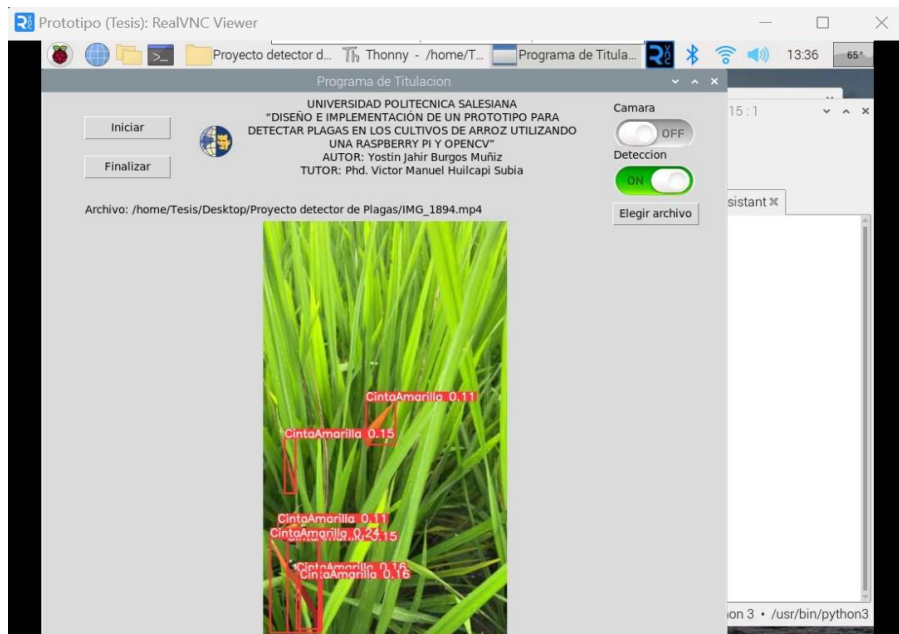


Figura 63 detección de plaga cinta amarilla

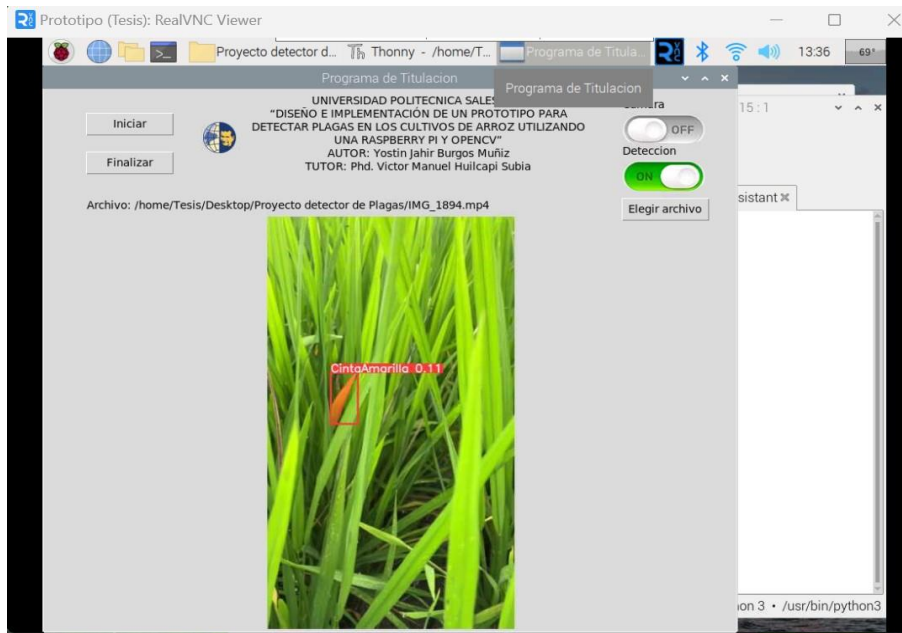


Figura 64 Captura detección de plaga cinta amarilla

VII CRONOGRAMA

El presente proyecto se pretende realizar en cuatro meses que comprende desde el mes de octubre del 2023 hasta febrero de 2024 las actividades que se realizarán se detallan a continuación:

Tabla 1 Cronograma de actividades

Actividades	Tiempo de duración																
	Meses	Octubre				Noviembre				Diciembre				Enero			
	semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Estudio																	
Compra de materiales (Raspberry, cámara, baterías, cables, conectores, etc.)																	
Ensamble del prototipo																	
Captura de diversas muestras de imágenes de las plagas (Enrolladora, mariposa blanca, cinta amarilla)																	
Programación																	
prueba del prototipo																	
Prototipo final																	
Presentación de proyecto																	

VIII PRESUPUESTO

Para la implementación del proyecto se necesitará comprar materiales, por lo tanto, se realizó un presupuesto detallado de la siguiente manera:

Tabla 2 Presupuesto para la implementación del proyecto

Unidad	Materiales	Precio Unitario	Presupuesto
1	Raspberry Pi	\$100.00	\$110.00
1	Cámara OV5647	\$20.00	\$20.00
1	Fuente de alimentación para Raspberry pi 4	\$10	\$10.00
1	Herramientas (caja para prototipo, cable de ret, microSD, cables y conectores, etc.)	\$80.00	\$80.00
1	Router	\$30.00	\$30.00
	TOTAL		\$250.00

IX CONCLUSIONES

La detección de plagas en los cultivos de arroz es crucial para garantizar la salud y rendimiento de las cosechas. La implementación de tecnologías avanzadas, como sistemas de monitoreo remoto y análisis de imágenes, ha mejorado significativamente la capacidad de identificar y gestionar problemas relacionados con plagas de manera temprana. Estas herramientas permiten una respuesta que va su vez ayuda a minimizar los daños causados por las plagas y a optimizar los esfuerzos de control.

Además, la detección de plagas en cultivos de arroz no solo se limita a la identificación visual, sino que también involucra técnicas como el uso de sensores y algoritmos de aprendizaje automático. Estas metodologías mejoran la detección porque contribuyen a una gestión más sostenible al reducir la dependencia de productos químicos que se utilizan en los cultivos de arroz.

En conclusión, el diseño del prototipo de detección de plagas como la cinta amarilla y mariposa blanca, se conectó una cámara a una raspberry pi 4 en la que se encarga de adquirir muestras de imágenes en donde se obtuvo muestras negativas y muestras positivas que sirvió para el entrenamiento del modelo con presencia y ausencia de las plagas para los diferentes escenarios que fueron evaluados, el prototipo funciona adecuadamente pero se debe tener en cuenta que al momento de detectar plagas hay factores que influyen mucho en esta como la luz, oscuridad y el entorno donde se encuentre. La detección en plagas se da de manera un poco lenta ya que la tarjeta raspberry pi es un ordenador que nos sirve para muchos proyectos, pero se requiere componentes con mayores recursos cuando se trata de visión artificial.

X RECOMENDACIONES

Proporcionar capacitación a agricultores sobre las nuevas tecnologías y métodos de detección. La concienciación sobre la importancia de la detección temprana puede promover la adopción de prácticas tecnológicas.

Implementar un sistema de alerta temprana que notifique a los agricultores sobre la presencia temprana de plagas

Considerar la integración de sensores adicionales para recopilar datos ambientales, como la humedad y temperatura, que puedan afectar la propagación de plagas.

Utilizar sensores remotos y dispositivos de Internet, para monitorear constantemente las condiciones del cultivo, incluyendo factores climáticos y de suelo. La integración de estos datos en plataformas de análisis puede mejorar la capacidad y detectar problemas de plagas.

La tecnología de drones autónomos equipados con cámaras y sensores puede proporcionar una visión aérea detallada de los campos de cultivo. Estos dispositivos permiten una cobertura para la detección temprana de áreas afectadas.

Crear plataformas que integren múltiples fuentes de datos, como imágenes satelitales, datos meteorológicos y análisis de suelo. Esto proporciona una visión holística y en tiempo real de la salud del cultivo.

XI REFERENCIAS BIBLIOGRÁFICAS

ADEPAP. (13 de Marzo de 2023). *Los métodos químicos para el control de plagas*.

Obtenido de Los métodos químicos para el control de plagas:

<https://adepap.cat/es/metodos-quimicos-control-de-plagas/>

Advanced IP Scanner. (2021). *Escanee una red en cuestión de segundos*. Obtenido de

Escanee una red en cuestión de segundos: <https://www.advanced-ip-scanner.com/es/>

AgroBase. (s.f.). *Novia del arroz/Rupela Albinella*. Obtenido de Novia del arroz/Rupela

Albinella: <https://agrobasesapp.com/panama/pest/novia-del-arroz>

Agrotterra. (08 de Junio de 2021). *Herbicidas, clasificación y uso*. Obtenido de Herbicidas,

clasificación y uso: <https://blog.agrotterra.com/descubrir/herbicidas-uso/77614/>

allSpares. (15 de Octubre de 2015). *¿Qué es una batería 18650?* Obtenido de ¿Qué es una

batería 18650?: <https://all-spares.com/es/articles-and-video/what-is-18650-battery/>

Alvarado, J. (26 de Mayo de 2012). *Procesamiento y Análisis*. Obtenido de Procesamiento

y Análisis: <https://www.tec.ac.cr/sites/default/files/media/doc/paid.pdf>

AVG. (19 de Agosto de 2022). *¿Qué es un router?* Obtenido de ¿Qué es un router?:

<https://support.google.com/googlenest/answer/6274087?hl=es->

[419#:~:text=Un%20router%20es%20un%20dispositivo,de%20%C3%A1rea%20local%20\(LAN\).](https://support.google.com/googlenest/answer/6274087?hl=es-419#:~:text=Un%20router%20es%20un%20dispositivo,de%20%C3%A1rea%20local%20(LAN).)

AyushExel. (11 de Diciembre de 2023). *ultralytics YOLO*. Obtenido de ultralytics YOLO:

<https://docs.ultralytics.com/es/>

Calvo, L. (10 de Marzo de 2022). *¿Qué es una Raspberry PI y para qué sirve?* Obtenido de *¿Qué es una Raspberry PI y para qué sirve?:* <https://es.godaddy.com/blog/que-es-raspberry-pi/>

CornellUniversity. (2003). *Valoración de la calidad de la imagen* . Obtenido de Valoración de la calidad de la imagen : <http://preservationtutorial.library.cornell.edu/tutorial-spanish/quality/quality-03.html#:~:text=Los%20factores%20claves%20en%20la,tono%2C%20y%20la%20apariencia%20general.>

DeustoFormación. (29 de agosto de 2022). *Datasets y Dataframes en Big Data: ¿qué son?* Obtenido de Datasets y Dataframes en Big Data: ¿qué son?: <https://www.deustoformacion.com/blog/programacion-tic/que-son-datasets-dataframes-big-data>

FAO. (s.f.). *Efectos económicos de las plagas y enfermedades transfronterizas*. Obtenido de Efectos económicos de las plagas y enfermedades transfronterizas: <https://www.fao.org/3/x9800s/x9800s16.htm>

González, M. (Marzo de 2021). *Técnicas de procesamiento de imágenes*. Obtenido de Técnicas de procesamiento de imágenes: <https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/26835/1/GON21.pdf>

Google nest. (Agosto de 2022). *¿Qué es un router y cuál es su función?* Obtenido de *¿Qué es un router y cuál es su función?:* <https://www.avg.com/es/signal/what-is-a-router>

Higieneambiental. (14 de Enero de 2016). *Resistencia a los insecticidas: ¿Qué es? ¿Cómo prevenirla?* Obtenido de Resistencia a los insecticidas: ¿Qué es? ¿Cómo prevenirla?: <https://higieneambiental.com/control-de-plagas/resistencia-a-los-insecticidas-que-es-como-prevenirla>

IEOFB. (7 de Enero de 2020). *¿En qué consiste un control de plagas?* Obtenido de ¿En qué consiste un control de plagas?: <https://ieqfb.com/control-de-plagas/#:~:text=La%20fumigaci%C3%B3n%20es%20uno%20de,se%20encuentran%20de%20forma%20gaseosa.>

Juarez, U. A. (01 de Agosto de 2023). *Introducción a las plagas.* Obtenido de Introducción a las plagas: <https://www.uacj.mx/ICB/UEB/documentos/1.%20PLAGAS.pdf>

Kubiliute, G. (28 de Junio de 2023). *La inteligencia artificial y su papel en la adaptación al cambio climático* . Obtenido de La inteligencia artificial y su papel en la adaptación al cambio climático : <https://www.ainia.es/ainia-news/inteligencia-artificial-herramienta-desafios-agricultura/#:~:text=AI%20aprovechar%20la%20IA%2C%20los,anticiparse%20a%20los%20cambios%20clim%C3%A1ticos.&text=La%20IA%20puede%20analizar%20grandes,cultivo%2C%20para%20generar%2>

Londoño, P. (4 de Abril de 2023). *Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender).* Obtenido de Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender): <https://blog.hubspot.es/website/que-es-python>

Luz, S. (03 de Junio de 2021). *Advanced IP Scanner: analiza tu red y los equipos conectados en Windows*. Obtenido de Advanced IP Scanner: analiza tu red y los equipos conectados en Windows:

<https://www.redeszone.net/analisis/software/advanced-ip-scanner-analizar-red-equipos-windows/>

Martín, L. (06 de Julio de 2020). *Tecnología autónoma para la detección real de plagas*. Obtenido de Tecnología autónoma para la detección real de plagas:

<https://www.tecnologiahorticola.com/tecnologia-autonoma-deteccion-plagas/>

NetApp. (2023). *¿Qué es la inteligencia artificial?* Obtenido de ¿Qué es la inteligencia artificial?: <https://www.netapp.com/es/artificial-intelligence/what-is-artificial-intelligence/>

NexusIntegra. (s.f.). *Ventajas y Desventajas de la Inteligencia Artificial en Empresas*.

Obtenido de Ventajas y Desventajas de la Inteligencia Artificial en Empresas:

<https://nexusintegra.io/es/ventajas-y-desventajas-de-la-inteligencia-artificial/>

Plantix. (2023). *Virus del moteado amarillo del arroz*. Obtenido de Virus del moteado amarillo del arroz: <https://plantix.net/es/library/plant-diseases/200021/rice-yellow-mottle-virus/>

Plantix. (s.f.). *Virus del moteado amarillo del arroz*. Obtenido de Virus del moteado amarillo del arroz: <https://plantix.net/es/library/plant-diseases/200021/rice-yellow-mottle-virus/>

Pública, I. N. (26 de Agosto de 2020). *Los insecticidas*. Obtenido de Los insecticidas:

<https://www.insp.mx/avisos/4736-insecticidas.html>

Rodríguez, H. (28 de Abril de 2021). *¿Qué es OpenCV?: ¡Descubre todo acerca de la visión artificial!* Obtenido de ¿Qué es OpenCV?: ¡Descubre todo acerca de la visión artificial!: <https://www.crehana.com/blog/transformacion-digital/que-es-opencv/>

Soto, J. (16 de Agosto de 2020). *¿Qué es VNC y para qué sirve?* Obtenido de ¿Qué es VNC y para qué sirve?: <https://www.geeknetic.es/VNC/que-es-y-para-que-sirve>

Tecnosoluciones. (18 de Octubre de 2019). *Fungicidas: Impacto en la Salud y el Medio Ambiente*. Obtenido de Fungicidas: Impacto en la Salud y el Medio Ambiente: <https://tecnosolucionescr.net/blog/145-fungicidas-impacto-en-la-salud-y-el-medio-ambiente#:~:text=Los%20fungicidas%20son%20sustancias%20t%C3%B3xicas,las%20plantas%2C%20o%20los%20animales.>

UNITElectronics. (2023). *Camara vision noctuena Raspberry pi4*. Obtenido de Camara vision noctuena Raspberry pi4: <https://uelectronics.com/producto/camara-vision-nocturna-raspberry-pi-5mp-1080p-ov5647/>

Uso de YOLO. (11 de Julio de 2023). *Entrenamiento de un modelo de detección de objetos con TensorFlow*. Obtenido de Entrenamiento de un modelo de detección de objetos con TensorFlow: <https://learn.microsoft.com/es-es/windows/ai/windows-ml/tutorials/tensorflow-train-model>

XII ANEXOS

PROGRAMACIÓN DE DETECCIÓN

```
# Importamos las librerías
```

```
from ultralytics import YOLO
```

```
from tkinter import *
```

```
from tkinter import filedialog
```

```
from PIL import Image, ImageTk
```

```
import imutils
```

```
import cv2
```

```
import numpy as np
```

```
import pandas as pd
```

```
def elegir_video():
```

```
    # Especificar los tipos de archivos, para elegir solo a las imágenes
```

```
    global path_file, archivoText
```

```
    path_file = filedialog.askopenfilename(filetypes = [
```

```
        ("video", ".mp4"),
```

```
        ("video", ".avi")])
```

```
    if len(path_file) > 0:
```

```
        textoFile.config(text = "Archivo: " + path_file)
```

```
# Funcion iniciar
```

```
def iniciar():
```

```

global cap

# Elegimos fuente de video

if encendidoCam:

    cap = cv2.VideoCapture(0)

else:

    cap = cv2.VideoCapture(path_file)

if not encendidoCam and not len(path_file) > 0:

    textoFile.config(text = "Escoja archivo Primero")

elif not encendidoCam and len(path_file) > 0:

    visualizar()

elif encendidoCam:

    visualizar()

# Funcion finalizar

def finalizar():

    cap.release()

    print("FIN")

# Funcion Visualizar

def visualizar():

    global cap, pantalla, frame, encendido, count, resultados, Alto1, Ancho1, Alto2,
Ancho2

    # Leemos la videocaptura

```

```

if cap is not None:

    ret, frame = cap.read()

    # Si es correcta

    if ret == True:

        frame = imutils.resize(frame, width=640)

        Alto1, Ancho1 = frame.shape[:2]

#    print(Ancho1 ,Alto1)

    if encendido:

        resultados = model.predict(frame, imgsz = 640, conf = 0.10)

        boxes = resultados[0].boxes.xyxy.tolist()

        class_ids = resultados[0].boxes.cls.tolist()

        class_names = resultados[0].names

        confidences = resultados[0].boxes.conf.tolist()

        print(boxes)

        print(class_ids)

        print(class_names)

        print(confidences)

        # Mostramos resultados

        anotaciones = resultados[0].plot()

        anotaciones = cv2.cvtColor(anotaciones, cv2.COLOR_BGR2RGB)

        # Convertimos el video

```

```

if Alto1 > Ancho1 :
    anotaciones = imutils.resize(anotaciones, width=300)
else:
    if encendidoCam:
        anotaciones = imutils.resize(anotaciones, width=600)
    else:
        anotaciones = imutils.resize(anotaciones, width=750)
Alto2, Ancho2 = anotaciones.shape[:2]
im = Image.fromarray(anotaciones)
img = ImageTk.PhotoImage(image=im)
else:
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    if Alto1 > Ancho1 :
        frame = imutils.resize(frame, width=300)
    else:
        if encendidoCam:
            frame = imutils.resize(frame, width=600)
        else:
            frame = imutils.resize(frame, width=750)
    Alto2, Ancho2 = frame.shape[:2]
    im = Image.fromarray(frame)
    img = ImageTk.PhotoImage(image=im)

```

```

    # count = count + 1

    #print(count

    x0 = 420-(Ancho2/2)

    y0 = 430-(Alto2/2)

    # Mostramos en el GUI

    lblVideo.configure(image=img)

    lblVideo.image = img

    lblVideo.after(10, visualizar)

    lblVideo.place(x = x0, y = y0)

else:

    cap.release()

# Definir funcion de interruptor

def switch():

    global encendido

    if encendido:

        on_button.config(image = off)

        encendido = False

    else:

        on_button.config(image = on)

        encendido = True

def switchCam():

    global encendidoCam

```



```

if encendidoCam:

    on_buttonCam.config(image = off)

    encendidoCam = False

else:

    on_buttonCam.config(image = on)

    encendidoCam = True

encendido = False

encendidoCam = False

archivo = False

path_file = ""

count = 0

Alto1 = 0

Ancho1 = 0

ruta = '/home/Tesis/Desktop/Proyecto detector de Plagas/'

model = YOLO(ruta + 'ModeloPlagas.pt')

# Read the Image

logo = Image.open('/home/Tesis/Desktop/Proyecto detector de Plagas/Logo.png')

# Resize the image using resize() method

resize_logo = logo.resize((40, 40))

titulo = ""UNIVERSIDAD POLITECNICA SALESIANA
“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA
DETECTAR PLAGAS EN LOS CULTIVOS DE ARROZ UTILIZANDO

```

UNA RASPBERRY PI Y OPENCV”

AUTOR: Yostin Jahir Burgos Muñiz

TUTOR: Phd. Victor Manuel Huilcapi Subia''

Ventana Principal

Pantalla

pantalla = Tk()

pantalla.title("Programa de Titulacion")

pantalla.geometry("840x700") # Asignamos la dimension de la ventana

Interfaz

Titulo = Label(pantalla, text=titulo)

Titulo.place(x = 250, y = 5)

Logo

#Logo = PhotoImage(resize_logo)

log = ImageTk.PhotoImage(resize_logo)

LogoL = Label(pantalla,image = log, text = "Logo")

LogoL.place(x = 180, y = 35, relwidth = 0.08, relheight = 0.08)

botonIniciar = Button(pantalla, text="Iniciar", width = 10, command=iniciar)

botonIniciar.place(x=50, y=30)

botonFinalizar = Button(pantalla, text="Finalizar",width = 10, command=finalizar)

botonFinalizar.place(x=50, y=80)

Definir nuestras imagenes

on = PhotoImage(file =ruta + "on.png")

```
off = PhotoImage(file =ruta + "off.png")

texto1 = Label(pantalla, text="Camara")

texto1.place(x = 700, y = 10)

on_buttonCam = Button(pantalla, image =off, bd = 0, command = switchCam)

on_buttonCam.place(x=700, y=30)

texto2 = Label(pantalla, text="Deteccion")

texto2.place(x = 700, y = 70)

on_button = Button(pantalla, image =off, bd = 0, command = switch)

on_button.place(x=700, y=90)

btn = Button(pantalla, text="Elegir archivo", width=10, command=elegir_video)

btn.place(x=700, y=140)

textoFile = Label(pantalla, text= "Archivo: Sin Escoger")

textoFile.place(x = 50, y = 140)

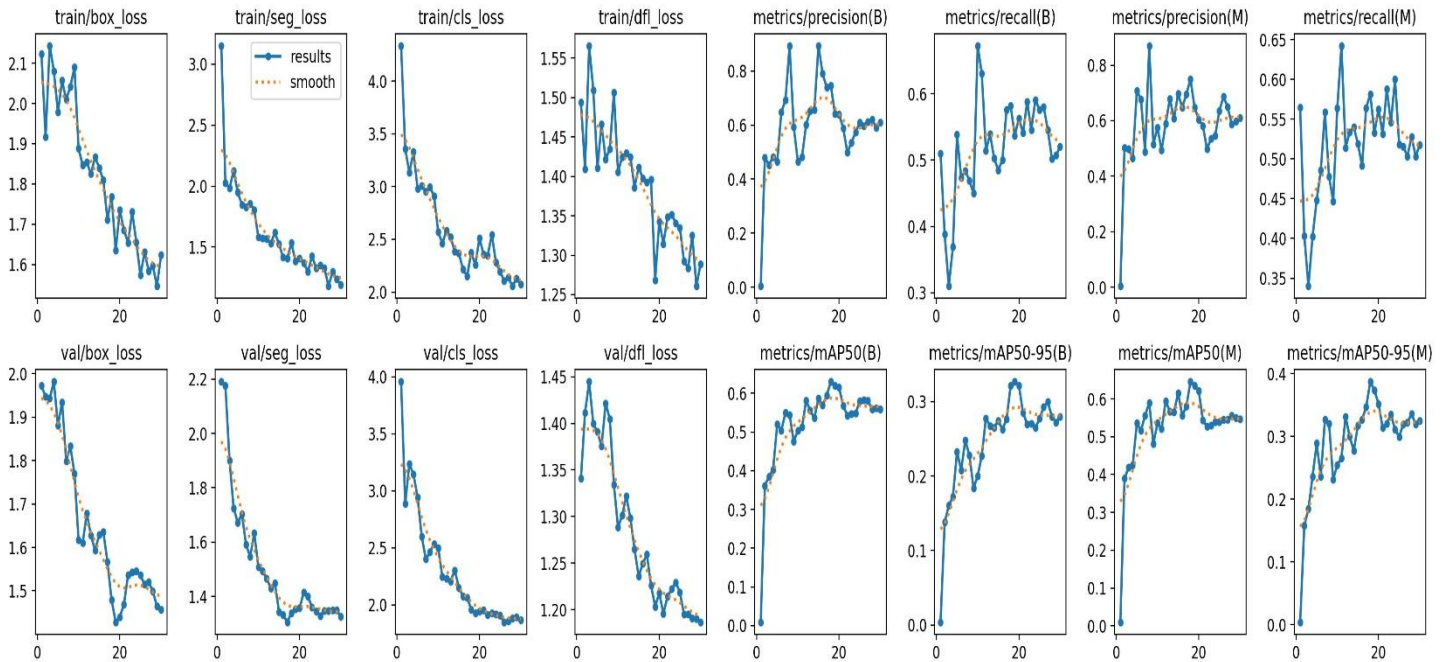
# Video

lblVideo = Label(pantalla)

pantalla.mainloop()
```

RESULTADO DE ÉPOCAS DEL MODELO ENTRANADO

epoch	train/box	train/seg	train/cls	train/df_l	metrics/pre	metrics/re	metrics/m	metrics/m#	metrics/pre	metrics/re	metrics/m	metrics/m#	val/box	val/seg	val/cls	val/df_l	lr/pg1	lr/pg2	lr/pg2
1	2.1227	3.1524	4.3371	1.494	0.00411	0.51002	0.00913	0.00388	0.00459	0.56446	0.0101	0.00404	1.9722	2.191	3.9578	1.3408	0.0005464	0.0005464	0.0005464
2	1.917	2.0284	3.3566	1.4096	0.47994	0.38795	0.36141	0.13795	0.50322	0.40351	0.38942	0.15768	1.9466	2.1773	2.8878	1.4115	0.0010657	0.0010657	0.0010657
3	2.1438	1.9826	3.1332	1.566	0.45272	0.3101	0.38514	0.16039	0.49747	0.34016	0.41857	0.18405	1.9427	1.8996	3.2352	1.4448	0.0015483	0.0015483	0.0015483
4	2.0809	2.1268	3.3274	1.5097	0.48293	0.36934	0.403	0.17236	0.46462	0.4024	0.42526	0.23629	1.9816	1.725	3.1462	1.3996	0.001502	0.001502	0.001502
5	1.9783	1.9491	2.9808	1.4104	0.46511	0.53833	0.52069	0.23284	0.70721	0.44817	0.5359	0.28871	1.8806	1.6715	2.9433	1.391	0.001502	0.001502	0.001502
6	2.0569	1.8489	3.0058	1.4661	0.64812	0.47343	0.50393	0.20804	0.67703	0.48563	0.51632	0.23608	1.9345	1.703	2.5998	1.3756	0.001447	0.001447	0.001447
7	2.0097	1.8302	2.9523	1.4218	0.69365	0.48389	0.55025	0.24806	0.48721	0.55891	0.55604	0.32644	1.7985	1.5908	2.4006	1.4213	0.0013919	0.0013919	0.0013919
8	2.0423	1.8585	2.9971	1.4347	0.89384	0.4689	0.54409	0.22768	0.8704	0.47762	0.59003	0.32051	1.8328	1.5466	2.4638	1.4048	0.0013369	0.0013369	0.0013369
9	2.0896	1.8039	2.9077	1.5063	0.59339	0.44991	0.47593	0.18378	0.51588	0.44685	0.48185	0.23121	1.77	1.6333	2.5354	1.3341	0.0012819	0.0012819	0.0012819
10	1.8881	1.5803	2.5686	1.4054	0.46603	0.67247	0.50317	0.19988	0.57409	0.56359	0.5356	0.25442	1.6167	1.5079	2.5013	1.2882	0.0012269	0.0012269	0.0012269
11	1.8464	1.5726	2.4626	1.4247	0.48181	0.63023	0.51213	0.22743	0.49358	0.64242	0.5212	0.26524	1.6104	1.4939	2.2441	1.3014	0.0011719	0.0011719	0.0011719
12	1.854	1.5877	2.5833	1.4296	0.60186	0.51394	0.58065	0.27761	0.5905	0.51394	0.59426	0.3314	1.6788	1.4666	2.2331	1.3218	0.0011169	0.0011169	0.0011169
13	1.8252	1.5318	2.5226	1.4245	0.65321	0.5392	0.55574	0.26722	0.67968	0.53354	0.56667	0.29969	1.6272	1.4303	2.2075	1.2981	0.0010619	0.0010619	0.0010619
14	1.8671	1.6187	2.384	1.3862	0.6567	0.50261	0.53571	0.26526	0.6077	0.53982	0.56528	0.27706	1.5933	1.4483	2.3038	1.2652	0.0010069	0.0010069	0.0010069
15	1.8397	1.5248	2.3676	1.4113	0.89454	0.48458	0.58673	0.2745	0.69643	0.51906	0.61466	0.31752	1.6292	1.344	2.1529	1.2357	0.0009519	0.0009519	0.0009519
16	1.8095	1.4162	2.2179	1.3977	0.79096	0.5006	0.56851	0.26241	0.64818	0.49129	0.55615	0.32648	1.6358	1.3316	2.0753	1.2496	0.0008969	0.0008969	0.0008969
17	1.7114	1.4098	2.15	1.3924	0.73905	0.57578	0.5949	0.27659	0.69515	0.56423	0.58047	0.34677	1.5688	1.3055	2.0651	1.2592	0.0008418	0.0008418	0.0008418
18	1.7689	1.5351	2.3719	1.3962	0.74711	0.58145	0.63004	0.3213	0.75	0.58145	0.64564	0.38722	1.4791	1.3395	1.9603	1.2261	0.0007868	0.0007868	0.0007868
19	1.6351	1.3835	2.2612	1.2685	0.64104	0.53699	0.62015	0.32682	0.64671	0.53267	0.63439	0.37334	1.4282	1.3529	1.9267	1.2034	0.0007318	0.0007318	0.0007318
20	1.7348	1.4056	2.5099	1.3429	0.63968	0.56272	0.61467	0.32172	0.60507	0.56272	0.62129	0.35166	1.4402	1.3569	1.9441	1.2178	0.0006768	0.0006768	0.0006768
21	1.6859	1.3692	2.3576	1.3146	0.5894	0.54068	0.56756	0.28494	0.58202	0.53151	0.54407	0.31379	1.469	1.4158	1.9561	1.1957	0.0006218	0.0006218	0.0006218
22	1.654	1.2992	2.3473	1.3484	0.4984	0.58798	0.54272	0.26986	0.4984	0.58798	0.52699	0.3202	1.5369	1.4002	1.9152	1.2142	0.0005668	0.0005668	0.0005668
23	1.7314	1.4258	2.5444	1.3516	0.53549	0.54573	0.54655	0.27078	0.53549	0.54573	0.53051	0.33518	1.5428	1.3607	1.9292	1.2221	0.0005118	0.0005118	0.0005118
24	1.6552	1.3282	2.2799	1.3408	0.57383	0.59063	0.54815	0.26467	0.54692	0.60017	0.53979	0.31074	1.5456	1.3461	1.9187	1.2293	0.0004568	0.0004568	0.0004568
25	1.5732	1.3499	2.1954	1.3346	0.60914	0.57578	0.57897	0.27722	0.63552	0.51823	0.53984	0.29951	1.5361	1.3283	1.9059	1.2188	0.0004018	0.0004018	0.0004018
26	1.6304	1.327	2.1068	1.2924	0.59652	0.58052	0.58257	0.2925	0.68638	0.51568	0.54602	0.31953	1.5146	1.3478	1.8493	1.1951	0.0003467	0.0003467	0.0003467
27	1.5837	1.1803	2.1386	1.2835	0.61272	0.54573	0.58086	0.29965	0.65036	0.50348	0.54509	0.32232	1.5205	1.3469	1.8555	1.1954	0.0002917	0.0002917	0.0002917
28	1.5973	1.2978	2.0565	1.3256	0.62097	0.50219	0.55916	0.27952	0.58657	0.52787	0.55613	0.3355	1.4988	1.3499	1.8828	1.1911	0.0002367	0.0002367	0.0002367
29	1.5464	1.2361	2.1268	1.2608	0.59177	0.50692	0.56195	0.27227	0.59694	0.50348	0.54973	0.31924	1.4649	1.3489	1.8931	1.1903	0.0001817	0.0001817	0.0001817
30	1.6232	1.1911	2.0713	1.2884	0.61035	0.51986	0.55938	0.27971	0.60932	0.51761	0.54643	0.32496	1.457	1.3256	1.8677	1.1867	0.0001267	0.0001267	0.0001267



CONFUSIÓN MATIX NORMALIZADO

LABELME CORRELOGRAMA

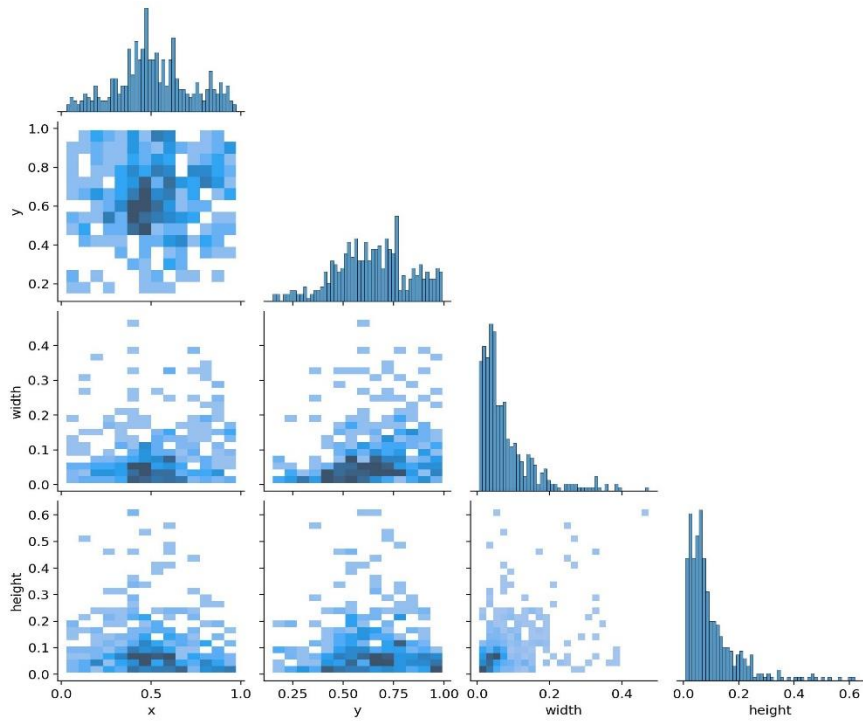
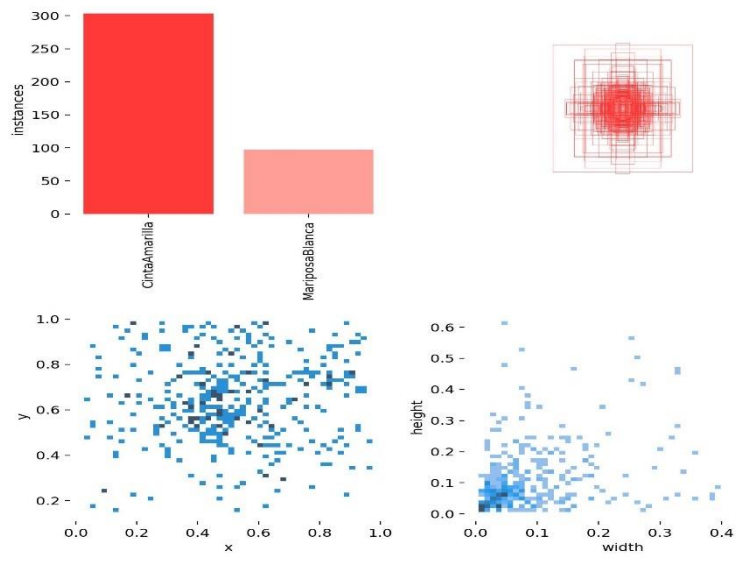


GRÁFICO LABELME

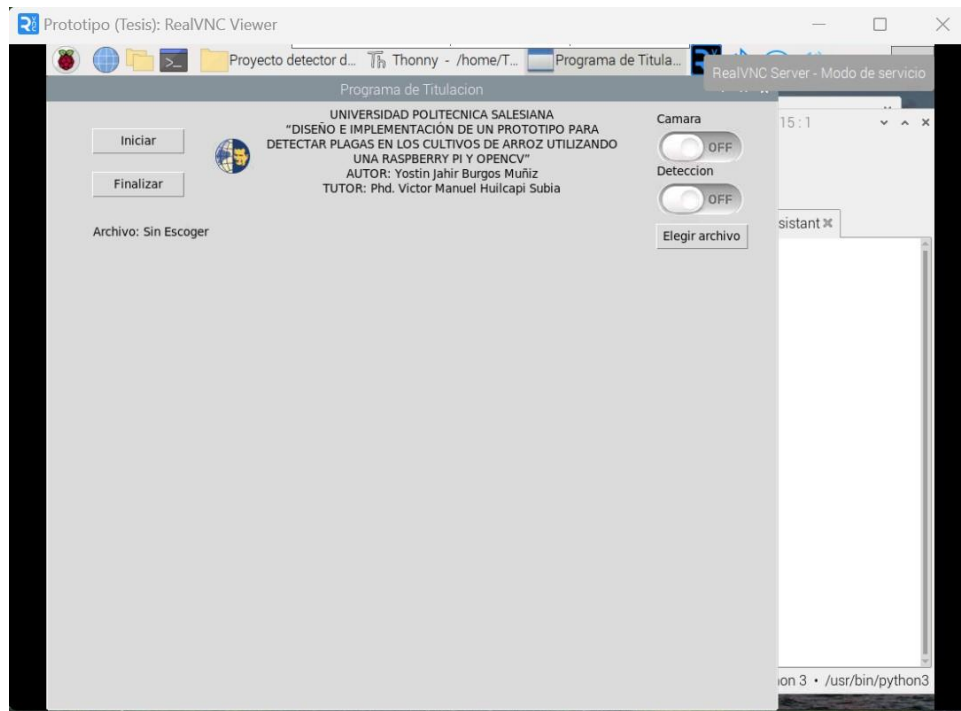


DETECCIÓN DE LAS PLAGAS POR EL PROTOTIPO





IMÁGEN DE LA INTERFAZ



PROTOTIPO FINAL

