UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TEMA: "VISIÓN POR COMPUTADOR PARA RECONOCIMIENTO DE MALEZAS EN CULTIVOS DE TOMATE RIÑÓN DE INVERNADERO, MEDIANTE REDES NEURONALES"

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTOR: KARLA FERNANDA MONCAYO SUÁREZ

DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA

Ibarra-Ecuador

2022

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

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

IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003194774		
APELLIDOS Y NOMBRES:	Moncayo Suárez Karla Fernanda		
DIRECCIÓN:	Los Ceibos, calle Río Orinoco 1-23 y Jubones		
EMAIL:	kfmoncayos@utn.edu.ec		
TELÉFONO FIJO:	062604315	**TELÉFONO MÓVIL:**	0992921324
DATOS DE LA OBRA			
TÍTULO:	"VISIÓN POR COMPUTADOR PARA RECONOCIMIENTO DE MALEZAS EN CULTIVOS DE TOMATE RIÑÓN DE INVERNADERO, MEDIANTE REDES NEURONALES"		
AUTOR (ES):	Karla Fernanda Moncayo Suárez		
FECHA: DD/MM/AAAA	10 de enero de 2022		
SOLO PARA TRABAJOS DE GRADO			
PROGRAMA:	☐ **PREGRADO** ☐ **POSGRADO**		
TITULO POR EL QUE OPTA:	Ingeniera en Electrónica y Redes de Comunicación		
ASESOR /DIRECTOR:	MsC. Edgar Alberto Maya Olalla		

CONSTANCIAS

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

Ibarra, a los 10 días del mes de enero de 2022

EL AUTOR:

Karla Fernanda Moncayo Suárez

CI: 100319477-4

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN.

MSC. EDGAR MAYA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN

CERTIFICA:

Que, el presente trabajo de Titulación "VISIÓN POR COMPUTADOR PARA RECONOCIMIENTO DE MALEZAS EN CULTIVOS DE TOMATE RIÑÓN DE INVERNADERO, MEDIANTE REDES NEURONALES". Ha sido desarrollado por la Srta. Moncayo Suárez Karla Fernanda, bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

.....

MsC. Edgar Maya.

DIRECTOR

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

El presente trabajo de titulación principalmente lo dedico a Dios, por ser mi sustento, bendecirme en cada paso de mi vida y permitirme el culminar este escalón, como lo es mi formación profesional.

A mi hijo Ethan Estrada, quien es el pilar más importante en mi vida; quien con su cariño y amor me ha dado las ganas de superarme cada día y ser una mujer profesional; quien, con su confianza en mí, ha logrado que saque las fuerzas necesarias para luchar y alcanzar todo lo que me proponga.

A mis padres, quienes con su apoyo incondicional me han acompañado paso a paso en este largo camino.

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTOS

Quiero agradecer a mi Madre Marthita por todo el amor, cariño y valores que me ha sabido inculcar; quien ha sido un ejemplo de lucha para mí. A mi papá Carlos por estar conmigo incondicionalmente. A mis hermanas Sandra y Brigitte y mi hermano Cristopher, quienes con sus consejos me han dado mucha fortaleza. A mi mejor amigo Andresito, quien me ha brindado su apoyo incondicional y siempre ha estado en los momentos más importantes de mi vida.

A mis amigas, compañeros, familiares y docentes de la FICA quienes han estado presentes en este proceso importante como lo es mi carrera profesional. De manera especial agradezco a mi director de tesis el MsC. Edgar Maya por todos y cada uno de sus consejos, sobre todo agradezco por las enseñanzas que ha sabido compartir conmigo durante toda la carrera.

Karla Fernanda Moncayo Suárez

Índice de Contenidos

Índice de Figuras

Índice de Tablas

Índice de Ecuaciones

ANEXOS

RESUMEN

En el presente trabajo de titulación se realiza un sistema de reconocimiento de malezas de tomate riñón de invernadero mediante redes neuronales; ya que en la actualidad no existen sistemas tecnológicos que ayuden a resolver problemas en dichos cultivos, causando enfermedades en el fruto por la detección tardía de las malezas.

Para su desarrollo, se ha considerado realizar un sistema de reconocimiento de visión por computador mediante redes neuronales; el mismo que se inicia con la adquisición de datos, se procede a etiquetarlos mediante el Software LabelImg. A continuación, se realiza el entrenamiento del sistema mediante redes neuronales convolucionales utilizando un lenguaje de programación, una vez culminado el entrenamiento se dispone del sistema deseado.

Con la utilización de un dispositivo embebido interactuando con una pantalla para visualizar el resultado, se dispone de un sistema en Python con el que se logra realizar las pruebas de funcionamiento; con las que se verifica el correcto funcionamiento de este. Las pruebas se las realiza con imágenes, videos guardados o en tiempo real.

ABSTRACT

In this study, neural networks are used to create a system for weed recognition in greenhouse tomato crops. Currently, there are no technological systems that can assist in the resolution of problems in such crops, eventually causing diseases in the fruit as a result of late weed detection.

For its development, it has been considered to make a computer vision recognition system through neural networks; the same one that begins with the acquisition of data proceeds to label them using the labelImg Software. The system is trained using convolutional neural networks and a programming language, and the desired system is made available once this process is completed.

There is a Python system that can be used to perform performance tests on an embedded device that interacts with a screen to visualize the result; this is how its correct functioning is verified. The tests are carried out with images, videos that have been saved or recorded in real-time.

CAPÍTULO I. ANTECEDENTES

En el capítulo mencionado, se encuentra detallado el trabajo de titulación a realizarse tales como: tema, problema, objetivos, alcance y justificación, con la finalidad de reconocer malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales", el cual ayudará en el reconocimiento de malezas en dichos cultivos.

1.1. Tema

"VISIÓN POR COMPUTADOR PARA RECONOCIMIENTO DE MALEZAS EN CULTIVOS DE TOMATE RIÑÓN DE INVERNADERO, MEDIANTE REDES NEURONALES".

1.2. Problema

La agricultura en el Ecuador ha sufrido diversos cambios, existe falta de recursos tecnológicos y aplicaciones que hacen que sea imposible explotar el verdadero potencial en el sector agrícola (Andrea et al., 2018). Con la presencia de malezas, se ve afectada la nutrición del cultivo de tomate riñón, y el poder detectar la misma, es difícil ya que ambos tienen el color predominante(verde) (Zhang et al., 2018). Para realizar cambios y mejorar la producción de tomate riñón, no se dispone de innovaciones tecnológicas, es por ello que se podría optar por un método que incluya tecnología para el reconocimiento de malezas mediante el procesamiento de imágenes y la clasificación, usando redes neuronales; ya que es lo que en la actualidad se usa en la visión artificial, teniendo buenos resultados(Gómez, 2015)

El reconocimiento de maleza en los cultivos de tomate riñón, mediante un método tradicional; logra que se utilice excesivamente herbicidas, pesticidas químicos, etc. , lo cual resulta costoso para los agricultores y podría generar un impacto negativo sobre el medio ambiente y para la salud humana.(Andrea et al., 2018). Según datos de la FAO, en los cultivos agrícolas a nivel

mundial existen pérdidas en la producción de tomate riñón, debido a la presencia de las mismas; las cuales se encuentran entre un 20% y 40%. La magnitud en cuanto a pérdidas varía acorde a la región, de año a año y depende del cultivo que se esté tratando. Se puede mencionar que se tiene pérdidas en el cultivo de tomate riñón del 24.5% del cual el 5.4% se debe a malezas (Nesheim et al., 2015). La maleza en un cultivo causaría algunas afectaciones como: Albergar insectos dañinos a las plantas cultivables, aumentar los costos de operaciones por obstruir el proceso de cosecha, sus semillas contaminan la producción del cultivo, su presencia reduce la eficiencia de la fertilización, facilita la existencia y crecimiento de otras plagas y su genética puede resultar tóxica para los cultivos (Monsanto, 2017).

En la actualidad no se utiliza la innovación tecnológica para ayudar al agricultor a reconocer las malezas en cultivos de tomate riñón. Razón por la cual se pretende realizar el sistema de reconocimiento de malezas mediante visión por computador utilizando redes neuronales; que ayudaría con la automatización de procesos, mediante el tratamiento de imágenes de malezas con las del cultivo, entrenando al sistema previamente y logrando que se realice un reconocimiento de malezas confiable.

1.3. Objetivos

1.3.1. Objetivo general.

Diseñar un sistema de visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales

1.3.2. Objetivos específicos.

- Elaborar una fundamentación teórica acerca de reconocimiento de malezas utilizando visión artificial mediante redes neuronales.
- Determinar los parámetros idóneos para la realización del sistema, utilizando el modelo en V.
- Entrenar el sistema mediante redes neuronales a partir de información recolectada en algunos cultivos de tomate riñón de invernadero.
- Construir el sistema de visión por computador para el reconocimiento de malezas en cultivos de tomate riñón de invernadero
- Realizar pruebas de funcionamiento a escala de laboratorio para la corrección del sistema.

1.4. Alcance.

El proyecto tiene como finalidad diseñar un sistema de visión por computador para el reconocimiento de malezas en cultivos de tomate riñón mediante redes neuronales.

Para el proyecto se realizará la fundamentación teórica, todo en cuanto a Sistemas de visión artificial, reconocimiento de malezas en cultivos de tomate riñón, Redes Neuronales. Para poder determinar los parámetros idóneos en la realización del sistema; se obtendrá bases bibliográficas respecto a visión artificial aplicado a malezas en los cultivos de tomate riñón de invernadero mediante redes neuronales.

Para determinar tanto software como hardware del sistema se lo realizará mediante el modelo en V, ya que éste nos va a permitir, describir las diversas actividades y los resultados a mostrar en su desarrollo. Al elegir el modelo de cámara a utilizar, será acorde a las características

requeridas para el sistema de reconocimiento de malezas en el tomate riñón. Con la utilización de una cámara y Software libre interactuando conjuntamente con un sistema embebido, se podrá reconocer a la maleza del cultivo de tomate riñón.

Para el entrenamiento del sistema mediante redes neurales se utilizará información recolectada en diferentes cultivos de tomate riñón, en la provincia de Imbabura. Las redes neuronales son una herramienta de aprendizaje y procesamiento automático, la red es entrenada para clasificar patrones de datos mediante las fotografías antes mencionadas. En la construcción del sistema, se utilizará una cámara configurada en un sistema embebido, utilizando un Raspberry y Software libre, que asistirá con el reconocimiento de imágenes.

Al final se realizarán las pruebas de funcionamiento del sistema a escala de laboratorio; permitiendo reconocer la existencia de malezas en el cultivo de tomate riñón de invernadero, mediante redes neurales; en base a los resultados se podrá redactar las conclusiones y recomendaciones respectivas.

1.5. Justificación.

En la actualidad, los retos que tienen en el sector agrícola, es la necesidad que se tiene por desarrollar innovaciones tecnológicas que aporten y favorezcan el desarrollo y la competitividad en dicho sector (Gómez, 2015).

Según datos estadísticos del Ministerio de Agricultura y Ganadería(MAGAP), indica que: de una producción plantada de 1970 ha, existe una superficie cosechada de 1954 ha, teniendo 16 ha no cosechadas (Magap, 2017). Las pérdidas que existen en los cultivos de tomate riñón, debido a la presencia de maleza, se ve afectada, ya que la agricultura en el Ecuador tiene falta de recursos tecnológicos y aplicaciones que hacen que sea imposible explotar su verdadero potencial como productor agrícola (Andrea et al., 2018).

Basados en los datos expuestos, se ve la necesidad de disponer de procesos que innoven el área de la agricultura, en este caso el poder reconocer malezas dentro de un cultivo de tomate riñón; valiéndose de la utilización de la tecnología que hoy en día es una solución a diversos problemas presentados. En la actualidad el uso de las tecnologías ayuda a que las mismas converjan logrando resultados satisfactorios, con la ayuda de la ingeniería se podrá tener cultivos con mayor producción(Gómez, 2015)

El desarrollo e innovación tecnologías que ayuden a administrar adecuadamente los recursos naturales que posee nuestros países está dentro de los objetivos que persigue el Plan Nacional del Buen Vivir, permitiendo a las poblaciones de los sectores rurales utilizar tecnologías actuales para mejorar la productividad para vivir en armonía con la naturaleza y sociedad. Por esta razón es importante desarrollar tecnologías que beneficien al medio amiente y el hombre. (PNBV, 2013-2017)

En la Universidad Técnica del Norte, la carrera de Ingeniería en Electrónica y Redes de Comunicación tiene como visión formar ingenieros humanistas, líderes y emprendedores que tengan una responsabilidad social; que genere, fomente y ejecute procesos tecnológicos; basándose en lo expuesto al realizar y contribuir con la realización de un sistema de visión por computador para el reconocimiento de malezas en los cultivos de tomate riñón; contribuye al sector agrícola.(UTN, 2019)

CAPÍTULO II. MARCO TEÓRICO

En este capítulo se recopila la información bibliográfica necesaria para la realización del presente proyecto. Entre los componentes teóricos a ser analizados, se habla sobre la importancia de los cultivos de tomate riñón de invernadero, así como también se define, qué es una maleza y sus características; por otro lado, se estudia las diferentes malezas que existen dentro de dichos cultivos. Posteriormente de esto, se realiza un estudio sobre visión por computador y sus procesos. A continuación, se estudia todo acerca de redes neuronales, sus características, los modos de operación que se tiene, sus elementos básicos, su estructura, y las arquitecturas que se manejan dentro de las mismas. De tal manera que se estudia de manera óptima, los temas más importantes y se dispone de la información necesaria para la realización del sistema.

2.1. Cultivos de tomate Riñón.

El cultivo de tomate riñón es uno de los más importantes en invernadero, ya que es de consumo masivo, su producción y rentabilidad es muy alta; volviéndose significativo en la economía de todas las familias (Pichisaca, 2003). El tomate riñón de invernadero tiene características como su forma que es más o menos redondeada, su peso es muy variable considerando su variedad y el desarrollo del cultivo. También se puede mencionar que el color del fruto aún no maduro es verde y cuando madura tiene un color rojo, amarillo, pardo, es decir con diferentes tonalidades, etc., esto depende a la variedad de que se trate (Antonio, 2016).

En la Figura 1, se muestra una fotografía de tomates riñón de invernadero con las características antes mencionadas, como son: su forma y su color, que en este caso llevan el color de rojo y verde.

Figura 1. Cultivo de Tomate Riñón de invernadero.

Fuente: (Productor, 2018)

2.1.1. Malezas.

(Ortega, 2015), menciona que las malezas son plantas que tienden a crecer en los cultivos agrícolas, los mismos que perjudican a dichas cosechas; también tienden a interferir en los cultivos de tomate; debido a que se encuentran en una competencia ya sea por luz, agua y nutrimentos del suelo. Por ello, se puede apreciar algunas características de las malezas, las cuales se debe comprender para conocer los tipos que existen dentro de los mencionados cultivos (AgroJornada, 2018).

Entre las características más destacadas se habla de: alta agresividad competitiva; que se refiere a la habilidad que tiene la maleza para aprovechar y almacenar los elementos vitales como: el agua, la luz, el dióxido de carbono y los nutrientes; los cuales se encuentran disponibles en la tierra. De igual manera, se dice que son altamente tolerantes a la variación ambiental y son plásticas, es decir, disponen de plasticidad en todos los tipos de malezas, lo que hace que sean más tolerables a las diversas condiciones y factores que se encuentren en el ambiente; de esta manera se genera un tipo de resistencia a los herbicidas. Así como también tienen un rápido crecimiento,

permitiendo alcanzar la fase reproductiva de forma temprana y aprovechar al máximo la floración en condiciones ambientales, las cuales se vuelven favorables para tener un crecimiento vegetativo continuo.

Una vez conocidas sus características; se puede mencionar más adelante acerca de los diferentes tipos de malezas que existen dentro de los cultivos de tomate riñón.

2.1.1.1. Tipos de malezas

(Flores, 2014) menciona que, dentro de los cultivos de tomate riñón, es necesario poder identificar algunos tipos de malezas; por lo que se dice que existen muchas variedades; entre ellas se puede mencionar:

- Malezas Anuales: son las que poseen un lapso de vida y crecimiento desde el primer momento en que se germinan hasta cuando ya se obtiene los frutos de un año, sus semillas pueden permanecer intactas en el suelo de 4 a 40 años.
- Malezas Perennes: tienen un ciclo de vida de dos a más años y son resistentes; su ciclo depende del ambiente si es o no favorable.
- Maleza Herbácea: las cuales tienen un comportamiento como el pasto y sus hojas son parecidas al mismo.
- Maleza de Hoja Ancha: se caracteriza por tener hojas amplias y planas, debido al tipo de hoja que tiene, son más factibles de separar en el momento que se encuentren pequeñas.

Adicionalmente, (Yujato, 2016) menciona que, es necesario conocer los tipos de malezas más problemáticas en el mundo, de las cuales, se puede destacar las siguientes:

- *Amaranthus dubius* – blero, , viene de la familia *amaranthaceae*, su propagación es directamente por las semillas y producen hasta 119 mil semillas/planta. Es una planta anual que tiene sus hojas alternas con peciolo largo y su inflorescencia es una panícula compuesta de espinas largas si son terminales o más cortas si son axilares. La *Amaranthus dubius* – blero se muestra en la Figura 2.

Figura 2. Amaranthus dubius - Blero

Fuente:(Soler Martínez Claudia, 2016)

- La maleza *Cyperus rotundus* – coquí, viene de la familia *Cyperaceae* y se reproduce por tubérculos y rizomas, es una planta perenne con un sistema radicular complejo compuesto de bulbos y es común en cultivos, pastizales y en gramas. Como se muestra en la Figura 3, su inflorescencia es una umbela terminal color café rojizo, sus hojas son alternas y se desarrollan en series de tres; también se puede decir que dicha planta es sensible a la sombra.

Figura 3. Cyperus rotundus - Coquí

Fuente: (Soler Martínez Claudia, 2016)

- Otra maleza que pertenece a la familia poaceae es la *Cynodon dactylon* - yerba bermuda; la cual es una planta perenne con tallo erecto y entrenudos alternos, uno corto y uno largo, como se muestra en la Figura 4. Dicha maleza, crece en terrenos baldíos, cultivos y potreros; su inflorescencia es solitaria y terminal con 4 a 6 espigas originadas en un mismo punto y se usa como césped y pasto.

Figura 4. Cynodon dactylon - Yerba Bermuda

Fuente: (Soler Martínez Claudia, 2016)

- Dentro de la familia *portulacaceae*, se encuentra la maleza portulaca oleracea – verdolaga, la cual es una planta anual que se reproduce por semillas y por fragmentos de tallo; dicha maleza es común en cultivos de hortalizas, ornamentales, y lugares abandonados. En la Figura 5, se muestra que su inflorescencia consta de cinco pétalos amarillos que sólo se abren en las mañanas soleadas; de igual manera sus hojas son alternas, frecuentemente están juntas al final de las ramas.

Figura 5. Portulaca oleracea - Verdolaga

Fuente:(Soler Martínez Claudia, 2016)

- La *Ipomoea* - Campanilla, es una maleza que ocasiona problemas de crecimiento en los cultivos. El crecimiento de esta maleza es de 5m anual. Dicha maleza prefiere suelos neutros y básicos, no puede crecer en la sombra, esta requiere de un suelo húmedo (Pardo & Saelzer, 2006). En la Figura 6 se muestra la forma de dicha maleza, teniendo las siguientes características: su raíz es fibrosa, su tallo es herbáceo con savia lechosa y sus hojas son grandes.

Figura 6. Ipomoea nil – -Campanilla

Fuente:(Aguirre et al., 2019)

- La *Sorghum halapense* - pasto Johnson, es una maleza que pertenece a la familia poaceae, la cual es una planta perenne, y se reproduce por rizomas y semillas; dicha maleza es común en cultivos anuales y perennes. Cabe mencionar, su inflorescencia es solitaria y terminal son en forma de pirámide. En la Figura 7 se muestra detalles como hojas planas, estrechas en la base y más anchas en el centro con orillas aserradas, que caracterizan a la *Sorghum halapense* - pasto Johnson.

Figura 7. Sorghum halapense - Pasto Johnson

Fuente:(Soler Martínez Claudia, 2016)

- La maleza *Digitaria sanguinalis* – pendejuelo, dicha maleza pertenece a la familia poaceae, la cual es una planta herbácea anual con tallos fornidos que se reproduce por semillas. Su inflorescencia es una panícula compuesta por varias espigas que parten de un mismo punto, parecida a la yerba pangola tal como se muestra en la Figura 8.

Figura 8. Digitaría sanguinalis - Pendejuelo

Fuente: (Soler Martínez Claudia, 2016)

- Por último, se tiene, la maleza *Rottboellia cochinchinensis* – caminadora, la misma que viene de la familia poaceae; ésta es una planta anual de tallo erecto que puede medir hasta tres metros de altura, como se muestra en la Figura 9. Se dice que su inflorescencia es terminal o axilar, y su espiga se hace más delgada hacia el ápice; dicha maleza está compuesta de entre nudos, cada uno conteniendo una semilla fértil.

Figura 9. Rottboellia cochinchinensis – Caminadora

Fuente: (Soler Martínez Claudia, 2016)

2.1.1.1.1. Malezas en el Ecuador

Las malezas en el Ecuador son aquellas plantas que causan reducciones en algunos cultivos, debido a la competencia de nutrientes que existe entre las mismas por el agua, la luz, el espacio, nutrientes, etc. De acuerdo con (Yujato, 2016). en el Ecuador las malezas que se mencionan son las siguientes: Marigold, Cadillo, Bledo, Verdolaga y Lechosa.

Cabe recalcar que, en determinados lugares, enfocándose en los dos primeros años de crecimiento de los cultivos, suelen atacar malezas que son bastante agresivas, como: Coquito (*Cyperus rotundus*), y Saboya (*Panicum maximun*).

(Zambrano, 2009) enfatiza que, al finalizar el estudio con las malezas en el Ecuador, es necesario conocer qué tipo de enfermedades se da en los cultivos de tomate riñón, es por ello por lo que mencionan las siguientes:

- Oidiopsis: en la que aparecen manchas amarillas en el haz de la parte superior de la hoja, y en el revés se observa una tela blanquecina, en ocasiones si el ataque ha sido muy fuerte, la hoja tiende a secarse y a caer.

- La podredumbre gris: es el tipo de parásito que ataca a las especies vegetales, con esto se ve afectado todo el cultivo; otra de las influencias en que haya dicha enfermedad de forma separada o continua es la temperatura, la humedad relativa y fenología (Zambrano, 2009).
- El *mildiu*: el cual es un hongo en el cultivo de tomate, que ataca específicamente a la parte aérea de la planta en cualquier etapa de su desarrollo; dicha enfermedad aparece como manchas con un aspecto aceitoso en sus principios, seguidamente invaden casi todo el foliolo.
- La *alternariosis*: esta enfermedad ataca principalmente al tomate y papa, en los cultivos se presenta tanto en hojas como tallos, frutos y pecíolos.
- El ataque de alternaria: comienza con la caída de pecíolos de hojas superiores, sus hojas inferiores se hacen amarillas y avanza hacia el ápice, ocasionando por último la muerte de la planta.
- El *verticilium dahliae kleb*: empieza por marchitar a la planta en horas de calor, seguidamente se da la clorosis en sus hojas. Finalmente, la planta termina marchitándose y muriendo, en muchos de los casos.

2.2. Visión por Computador.

La visión por computador provee soluciones reales y eficientes; que se logra al procesar, analizar e interpretar un conjunto secuencial de datos en una entrada ya sean estas de imágenes o de video (Pineda, 2018).

(Bariga, 2017) considera que, la visión artificial dentro de lo que es la inteligencia artificial; es como un conjunto de modelos y técnicas, las cuales permiten realizar un procesamiento, análisis y la explicación necesaria de cualquier tipo de información que se obtenga a través de imágenes

digitales. Dentro de lo que es el procesamiento de la imagen, se interpreta todas las características en secuencia, para incrementar su calidad y obtener una comprensión automática.

2.3. Procesamiento de Imágenes

El procesamiento de imágenes es un conjunto de técnicas que se aplican con la finalidad de facilitar la información buscada dentro de la misma (Bennett, 2014). Cabe recalcar que para el desarrollo de métodos del procesamiento, se basa en dos áreas de aplicación significativas: el mejoramiento de la información pictórica para la interpretación humana, y el procesamiento de datos de la imagen para la percepción de máquina autónoma (Andrés Montenegro, 2015).

(De la Escalera, 2017) menciona que, en la actualidad, los avances tecnológicos nos permiten obtener grandes cantidades de datos a través de dispositivos ya sea de imágenes por computador, escáner, teléfonos móviles o cámaras digitales; los mismos que pueden ser procesados para obtener información que nos sea útil en aplicaciones de visión artificial.

2.3.1. Representación de una Imagen.

A una imagen se la puede definir como una función de dos dimensiones f(x,y), como se muestra en la Figura 10, dicha imagen se la puede establecer como una matriz en los que las filas y columnas son los que asemejan un punto de la imagen y tienen el valor del elemento correspondiente (Calderón, 2012).

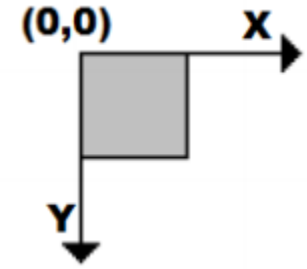

Figura 10. Representación de imágenes digitales

Fuente: (Calderón, 2012)

Además, se puede decir que, dentro de las imágenes, es necesario conocer sus propiedades; para ello se mencionan tres, tales como: La resolución espacial, que se trata del número de pixeles por unidad de longitud, ya que la imagen es una matriz X de dimensiones m x n pixeles, donde m son las filas y n las columnas. La resolución espacial está relacionada con aquellos detalles que se hacen visibles; el píxel representa el detalle más pequeño en una imagen.

También se tiene la resolución de niveles de escala de grises, que tiene que ver con la profundidad del píxel, el cual indica el número de bits por píxel en niveles de gris que se podrán apreciar en la imagen. Y como última característica, se tienen los planos de una imagen, que son los números de arreglos de pixeles que componen una imagen, por ejemplo: en una imagen en tonos de gris está compuesta de tres planos iguales, y una imagen de color estaría compuesta por tres planos distintos: sus componentes serían roja, la verde y la azul.

Una vez estudiadas las características de una imagen digital, cabe recalcar que; al tener una entrada, en este caso una imagen, se enlaza con el tema antes tratado como visión artificial, en la que al continuar el conjunto de procesos, se pueden obtener las características antes mencionadas e interpretar dichas imágenes (García, 2015).

Para ello, es necesario mencionar a continuación, el análisis que se debe realizar a una Imagen.

2.3.1.1. Análisis de una Imagen

Para poder analizar las imágenes mediante visión artificial se tiene 5 pasos fundamentales que son: a) La adquisición de imágenes, en la que, para poder adquirir una entrada, se necesita de un sensor y la capacidad de digitalizar la señal producida por el mismo. El segundo paso es, b) El preprocesamiento, ya que su función principal es mejorar la imagen para asegurar el éxito de los demás pasos; para ello existen técnicas de preprocesamiento como: el incremento del contraste, eliminación de ruido, o aislamiento de regiones. Una vez terminado este paso se realiza, c) La segmentación, que es, nada menos que la división de la imagen en sus partes constituyentes u objetos, su propósito consiste en dividir a la imagen en varias áreas significativas, de modo que el objeto puede ser analizado de mejor manera y se puede tener una interpretación óptima de datos. Luego se continua con el paso d) El reconocimiento e interpretación, el cual es el proceso que asigna una etiqueta a un objeto que contenga la información que se ha proporcionado anteriormente; es decir, en la interpretación se asigna el significado de objetos reconocidos. En este proceso es indispensable detallar las regiones de una imagen donde la información de interés se conoce para poder ser localizada, y la búsqueda arroje la información necesaria. Finalmente se tiene e) La representación, el mismo que es el último paso en el que se puede representar la imagen procesada; obteniendo una salida como resultado; una vez se hayan realizado todos los pasos anteriormente mencionados.

2.4. Aprendizaje Automático.

(Infaimon, 2018) describe que el aprendizaje automático dispone de grandes ventajas con respecto al descubrimiento de patrones mediante el uso de algoritmos matemáticos. Dicha técnica

se usa para poder clasificar las imágenes o para la toma de decisiones. El aprendizaje automático se divide en dos tipos principales, que son: supervisados y no supervisados.

2.4.1. Aprendizaje Supervisado.

Los algoritmos que se tiene en el aprendizaje supervisado, trabajan con datos; los cuales se encuentran etiquetados (Santos, 2017). Por ejemplo (Calvo, 2019) menciona que, la etiqueta es una salida que se muestra a partir del conjunto de datos ya conocidos. Por otro lado (Patricia & Arredondo, 2009) menciona que, al usar los datos que se tiene, se construye un modelo de predicción, el cual permite saber la clase para crear nuevos objetos que no se han visto aún. Los algoritmos más utilizados son: *Naive Bayes*, Máquinas de Vectores Soporte, Vecinos más cercanos, entre otros.

Dentro del aprendizaje supervisado se tiene la siguiente clasificación a) Aprendizaje Supervisado por regresión, como se visualiza en la Figura 11; en la cual se muestra que la regresión es un aprendizaje que predice un valor real, basado en las entradas; así como también se muestra los tipos de algoritmos que dispone esta clasificación por regresión. Dentro de la clasificación de aprendizaje supervisado, también se tiene b) Aprendizaje por clasificación, como se muestra en la Figura 12; y (González, 2018) sostiene que, este es un algoritmo que intenta etiquetar cada ejemplo eligiendo entre dos o más clases diferentes:.

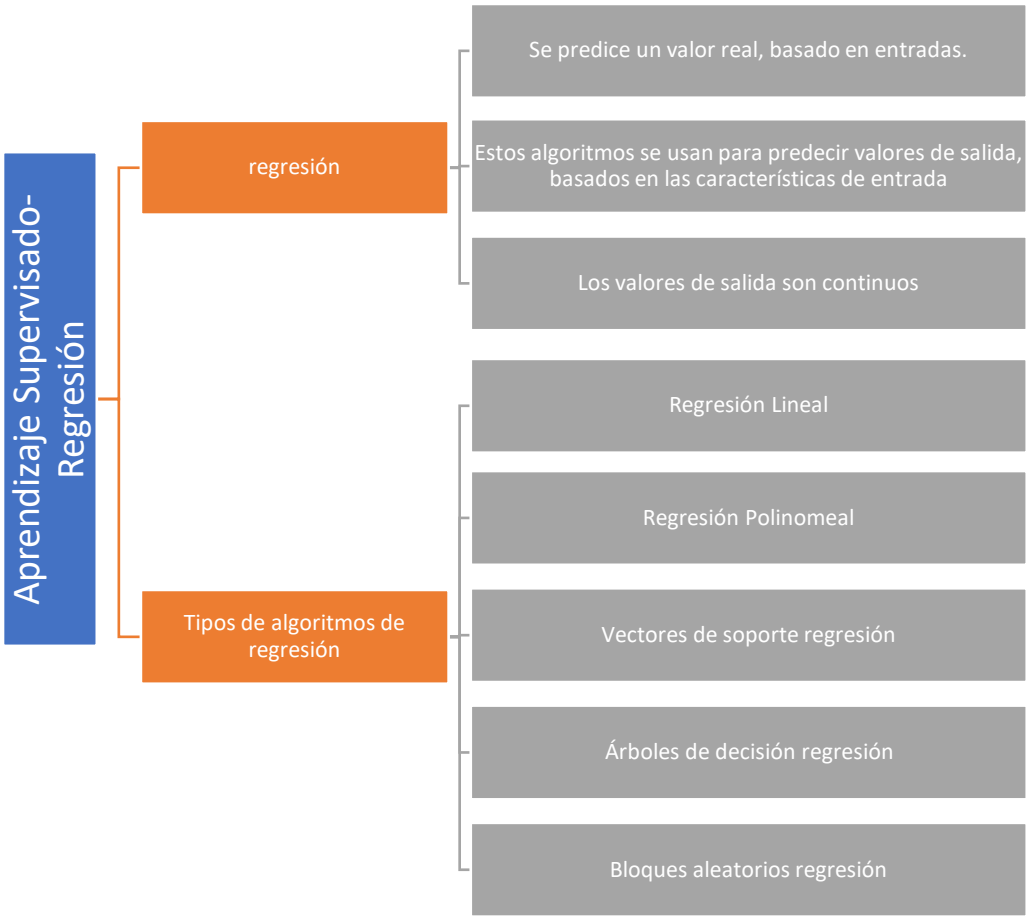

Figura 11. Aprendizaje Supervisado- Regresión

Fuente: Elaboración propia

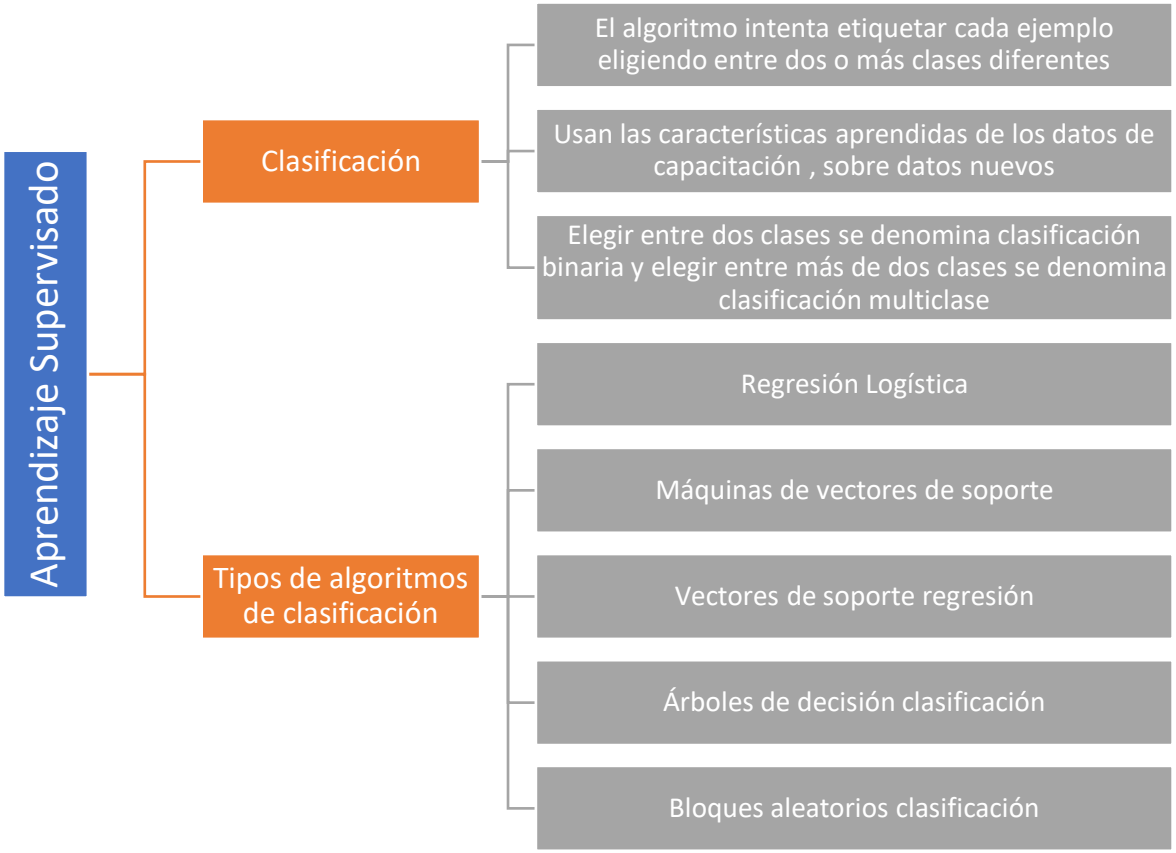

Figura 12. Aprendizaje Supervisado- Clasificación

Fuente: Elaboración propia

2.4.2. Aprendizaje no supervisado

El cual utiliza un conjunto de datos; los cuales no disponen de etiqueta alguna. La idea es que dicho algoritmo pueda encontrar patrones por sí sólo para entender el conjunto de datos (Alvarado, 2018). Este aprendizaje se usa generalmente, en problemas de clustering, agrupamientos y de *co-ocurrencia*. Los tipos de algoritmo más habituales son: a) Algoritmos de clustering, b) Análisis de componentes principales, c) Descomposición en valores singulares y d) Análisis de componentes independientes.

Al hablar del análisis de aprendizaje no supervisado, (Román, 2019) menciona, que es un proceso, el cual se lleva a cabo con los siguientes pasos: En la Figura 13, se establecen los pasos

secuencialmente; empezando por los datos, seguido de esto, se realiza una selección de sus características para pasar al perfeccionamiento del algoritmo de agrupación, una vez realizado este proceso se hace la validación de agrupación para pasar a la interpretación de resultados y finalmente se realiza la verificación del algoritmo de agrupación.

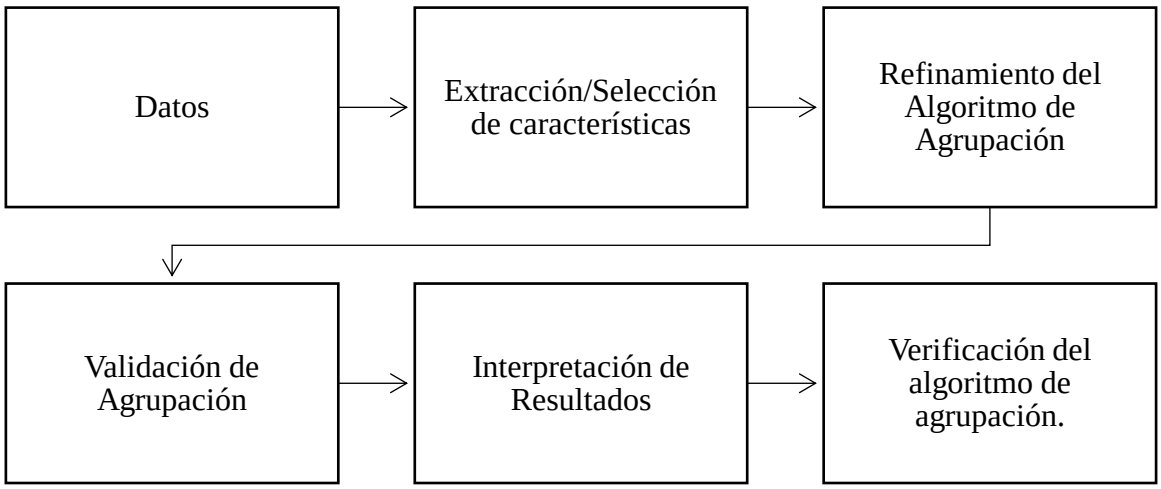

Figura 13. Proceso de Análisis de Aprendizaje no supervisado

Fuente: (Román, 2019)

2.5. Redes Neuronales.

La redes neuronales son un elemento primordial de las tecnologías de inteligencia artificial; son modelos matemáticos que son construidos a base del funcionamiento de las redes neuronales biológicas, se las conoce también como un conjunto de unidades de procesamiento a las cuales se las llama nodos, que están interconectadas entre sí por varias ligaduras; las cuales se comunican directamente, con la finalidad de que las mismas puedan recibir las señales de entrada, las puedan procesar y emitan la señal de salida. Las conexiones que las neuronas tienen se asocian

directamente con un peso, el cual contiene toda la información que se va a utilizar para poder resolver el problema(Vega, H., Cortez, A., Huayna, A., Loayza, L. y Naupari, 2009).

Las Redes Neuronales con el fin de parecerse al cerebro, tiene sus mismas características, las cuales se mencionan a continuación: Aprender, es una característica que tiene el poder adquirir ciertos conocimientos, por otro lado, en las redes neuronales se presenta un conjunto de entradas, las cuales se ajustan para procesar una salida. Otra característica es la de, generalizar, en las redes neuronales se generaliza gracias a su estructura y naturaleza; las mismas pueden ofrecer respuestas correctas a ciertas entradas en las cuales se presentan muy pequeñas variaciones ya sea por distorsión o ruido; y se tiene una última característica que es la de, abstraer, que significa considerar de manera separada cada una de las cualidades del objeto a procesar, algunas redes neuronales tienen la capacidad de abstraer las propiedades de los conjuntos de entradas en las que no se dispone de características muy comunes.

(Galán & Martínez, 2015) enfatiza que, las redes neuronales presentan en su mayoría las características de un cerebro humano; es por eso que se tiene las siguientes ventajas: Dispone de un aprendizaje Adaptativo, el cual posee la capacidad de aprender a realizar tareas que se encuentren basadas en una experiencia inicial. Las RNA (redes neuronales artificiales) tienen una autoorganización, lo cual quiere decir que es tan eficiente, que puede crear su representación a partir de la información que recibe por medio de un aprendizaje.

De igual manera es tolerante a fallos, ya que en algunas ocasiones se tiene la degradación de su estructura, a pesar de ello la red neuronal, posee la capacidad de retener las capacidades aún después de haber sufrido un daño. Otra de las ventajas que poseen las RNA es que su operación es en tiempo real; es decir, que con la ayuda del diseño de un hardware especifico se puede realizar cómputos neuronales. Cómo una última ventaja a mencionarse, se dice que es de fácil inserción

dentro de la tecnología existente, ya que es una ayuda para facilitar la integración modular de dichos sistemas, y se podría adquirir chips que sean netamente para redes neuronales; los cuales mejorarían notablemente su capacidad en ciertas áreas.

Una vez que se conocen las ventajas que disponen las redes neuronales, se debe conocer los elementos básicos de una red neuronal, más adelante se detalla dichos aspectos.

2.5.1. Elementos básicos de una red neuronal.

La red neuronal se encuentra constituida por neuronas interconectadas y sistematizadas en tres capas hasta n capas (Pablo & Toro, 2013). Los elementos básicos de una RNA, como se puede apreciar en la Figura 14; indica las capas que se tiene en una red neuronal totalmente conectada; la misma que, dispone de una capa de entrada en la que se ingresan los datos, los mismos que pasan por la capa oculta para arrojar los datos por la capa de salida.

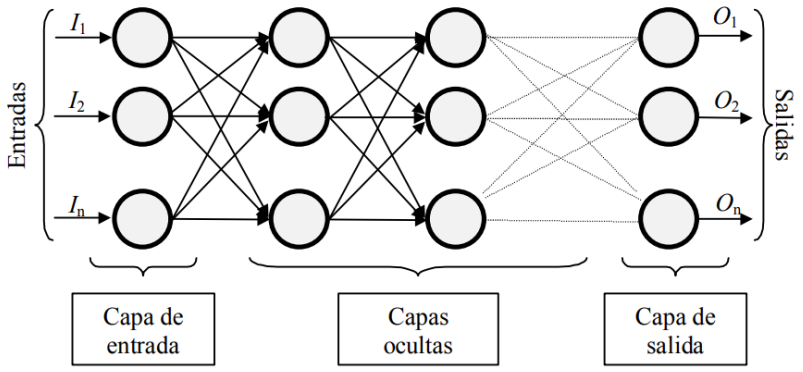

Figura 14. Ejemplo de una red neuronal totalmente conectada

Fuente: (Pablo & Toro, 2013)

Dentro de lo que es la capa de entrada, donde las neuronas tienen un tratamiento con los valores como si fueran uno sólo, por ello se la conoce como entrada global. Para poder combinar las entradas que se tiene (ini1, ini2, ...), siendo ini1= la entrada número 1 a la neurona Ni; la cual

se logra a través de la función de entrada y se calcula a partir del vector de entrada. La función de entrada como se muestra en la Ecuación 1; en la que,(Pablo & Toro, 2013), puede describir la función de entrada en donde los valores se multiplican por los pesos ingresados con anterioridad.

$$input_i = (in_{i1} \,.\, w_{i1}) * (in_{i2} \,.\, w_{i2}) * \dots (in_{in} \,.\, w_{in})$$

Ecuación 1

En donde se detalla que:

- El símbolo * representa al operador (máximo, sumatoria, producto, etc.),
- n representa al número de entradas a la neurona Ni
- Ni son las variables de la neurona y
- wi representa el peso correspondiente a in_{i1}, in_{in}

Seguido de esto los pesos que no tienen restricción, tienden a cambiar la influencia que representan en los valores de entrada. (Pablo & Toro, 2013) menciona que también se puede distinguir algunas funciones de entrada como: Sumatoria de las entradas pesadas, como se muestra en la Ecuación 2; la cual es la suma de los valores de entrada a la neurona, multiplicados por sus correspondientes pesos. Su ecuación quedaría de la siguiente manera:

$$\sum_j (n_{ij}\, w_{ij})\,, con\ j = 1, 2 \dots \dots \dots, n$$

Ecuación 2

Otra función es la, Productoria de las entradas pesadas, que viene a ser el producto de los valores de entrada a la neurona, indicándose en la Ecuación 3, los cuales son multiplicados por sus correspondientes pesos. Su ecuación quedaría de la siguiente manera:

$$\prod_{j}(n_{ij}w_{ij}),\ \ con\ j = 1, 2 \dots \dots \dots, n$$

Ecuación 3

Finalmente (Pablo & Toro, 2013) menciona la última función, se encuentra descrita en la Ecuación 4; en la que se tiene el máximo de las entradas pesadas, la cual, se toma en consideración solamente el valor de entrada más fuerte, previo a la multiplicación por el peso correspondiente. Su ecuación quedaría de la siguiente manera:

$$\text{Max}\ (n_{ij}w_{ij}),\ \ con\ j = 1, 2 \dots \dots \dots, n$$

j

Ecuación 4

Dentro de la función de activación, se encarga de devolver una salida a partir de un valor de entrada, generalmente sería un conjunto de valores de salida establecido en un rango de (0,1) o (-1,1) (Calvo, 2018).

En la Figura 15, se puede mencionar las siguientes funciones de activación. (Bootcamp AI, 2019)propone cada una de las siguientes funciones:

- Función Identidad, esta función permite que lo que se tiene en la entrada, sea igual a la salida; es decir, si se tiene una red neuronal de varias capas y se aplica una función identidad, se dice que es una regresión lineal. De tal manera que, si se aplica dicha función, se genera un valor único.
- Función Escalón, dicha función indica que, si x es menor que cero, la neurona resultante también es cero. Pero se tiene que, x es mayor o igual a cero, la neurona resultante tendrá una salida igual 1.

- Función sigmoidea: Esta función se encuentra en un rango en los que la salida está entre cero y uno (probabilidad), si se tiene valores de entrada negativos, su función es igual a cero.

Función	Formula	Rango
Identidad	$y = x$	$[-\infty,\infty]$
Escalón	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$	[0,1]
	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$	[-1,1]
Lineal a tramos	$y = \begin{cases} x & \text{si } -1 \leq x \leq 1 \\ +1 & \text{si } x > 1 \\ -1 & \text{si } x < -1 \end{cases}$	[-1,1]
Sigmoidea	$y = \frac{1}{1+e^{-x}}$	[0,1]
	$y = \tanh(x)$	[-1,1]
Sinusoidal	$y = Sen(\omega x + \varphi)$	[-1,1]

Figura 15. Funciones de activación

Fuente: (RNeuronales, 2012)

Finalmente se tiene la función de salida, el valor que se obtiene es la salida de la neurona i(out); es decir la salida es la que determina que valor es la que se transfiere a todas las neuronas que se encuentran vinculadas. Los valores de salida se encuentren en el rango [0, 1] o [-1, 1]; o de igual manera pueden ser binarios {0, 1} o {-1, 1}. En las funciones de salida se tiene dos funciones comunes tales como: a) Ninguna, si la salida es la misma que la entrada, conocida también como función identidad; y b) como se muestra en la Ecuación 5, es una ecuación binaria, la cual tiene parámetros de 0 y 1.

$$\begin{matrix} a) \\ b) \end{matrix} \begin{cases} 1 & si\ act_i \geq \varepsilon_i \\ 0 & de\ lo\ contrario \end{cases}, \qquad donde\ \varepsilon_i\ es\ el\ umbral \qquad \text{Ecuación 5}$$

Una vez aprendidos los elementos básicos de una red neuronal; se conoce la estructura que tiene un RNA; ya que su funcionamiento será similar al tema anteriormente revisado

2.5.2. Estructura de las redes neuronales.

Para describir la estructura de una red neuronal, primeramente, se debe tener claro la definición de neurona. (Galán & Martínez, 2015) describe a una neurona como la unidad básica de la red; que se la puede describir al compararla con una neurona biológica, ya que su funcionamiento será similar.

Una neurona biológica, la cual ésta formada por sinapsis, axón, dendritas y cuerpo mientras que una neurona artificial, es una unidad de procesamiento de información, un dispositivo de cálculo, el cual está formado por un vector de entradas que proporcionan una única salida.

Cabe mencionar que dentro de la estructura que posee un RNA; existen algunas arquitecturas con las que trabajan las redes neuronales; de las cuales se estudia la más utilizada como es: la arquitectura de *Yolo*, dicha arquitectura se detalla a continuación.

2.5.2.1. Arquitectura Yolo.

La Arquitectura Yolo se refiere a la detección de objetos, que consiste en una red neuronal convolucional que predice múltiples cuadros delimitadores y las probabilidades de la clase de objeto que delimitan dichos cuadros delimitadores.

(Enrique, 2018) menciona que en la arquitectura Yolo, para la detección de objetos se utiliza aprendizaje profundo y CNN; lo cual permite detectar fácilmente objetos en tiempo real. Así como también (Massiris Manlio , Juan Álvaro Fernández Muñoz, 2018) sostiene que, Yolo predice cada cuadro que delimita los objetos enmarcados simultáneamente, dando resultados por cada clase.

(Guti, 2019) enfatiza que, la arquitectura de la red neuronal que utiliza Yolo se encuentra formada por capas convolucionales seguida por dos capas de tipo *fully connected*; las cuales se las puede observar en la Figura 16.

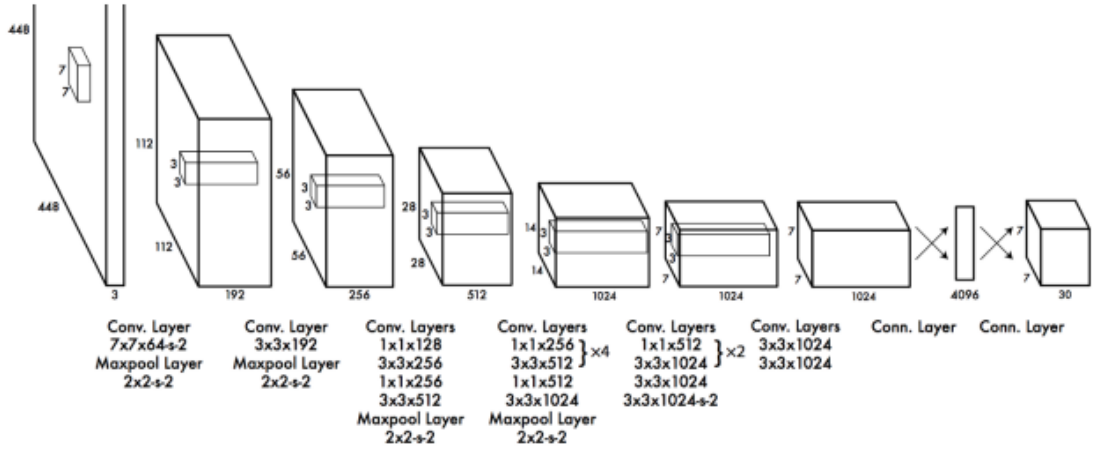

Figura 16. Arquitectura YOLO

Fuente: (Guti, 2019)

Cabe mencionar para mayor entendimiento de la arquitectura Yolo un modelo explicativo, el cual se encuentra basado en el dataset Pascal VOC (*Visual Object Classes*), la salida de este modelo es de un tamaño 7x7x30; para lo cual se especifica que en este tensor[1] se codifica 2 definiciones de recuadro y 20 probabilidades de clase en donde:

- S: es una dimensión espacial del tensor (7 en este caso)
- B: es el número de cuadros delimitadores (x, y, w, h, confianza) y
- C: número de clases

Se dice que 7X7 representa una cuadrícula de tales dimensiones y representa la imagen de entrada; donde cada celda de este tensor tendrá las definiciones de 2 cuadros y 20 probabilidades de clase.

Entre los beneficios y desventajas del algoritmo Yolo, se pueden mencionar las siguientes: Yolo obtiene buenos resultados rápidamente en tiempo real, de igual manera realiza las

[1] Sistema formado por magnitudes de una misma índole y que coexisten. Un tensor puede representarse como una matriz.

predicciones en una sola red neuronal y se puede entrenar de principio a fin, para mejorar la precisión, también se dice que Yolo accede a toda el área de la imagen para realizar predicciones y se menciona que Yolo detecta un objeto por celda. Al igual que se menciona ventajas, Yolo dispone de una desventaja que es: en el caso de que se tenga una sola celda, Yolo sólo detecta un solo objeto; lo cual limita lo cerca que los objetos pueden llegar a estar, ya que si la proximidad es elevada Yolo no será capaz de detectar todos los objetos.

De igual manera se pueden considerar las siguientes mejoras para Yolo v2: dispone de normalización de lotes: esto se añade en las capas convolucionales, y se elimina la necesidad de aplicar *dropout*, de igual manera se desactiva algunas neuronas aleatoriamente con el fin de evitar que se sobrecaliente. Así como también tiene un clasificador de alta resolución, que significa que al entrenar Yolo, se tiene dos fases; la fase uno en la que se entrena una red de clasificación y la fase dos en la que se reemplaza las capas fully connected por capas convolucionales y se las vuelve a entrenar para la detección de objetos; otra de las mejoras que se encuentra en el anchor boxes**,** es que en Yolo v2 se tiene la eliminación de las capas fully connected, que son las responsables de la predicción del *Bounding box*(BB), se añade alguna capa convolucional y se modifica la dimensión de alguna capa. De esta manera las predicciones se las realizará en base a los anchors.

Una vez, que se conoce la arquitectura Yolo; se realiza la investigación de la arquitectura CNN (red neuronal convolucional), la cual es una red multicapa jerárquica que extrae características en las capas que tiene ocultas

2.5.2.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) se utilizan para el procesamiento de imágenes. Mediante estas redes neuronales se elimina la necesidad de realizar extracciones

manuales de características, por lo que ya no sería necesario identificar las características utilizadas para clasificar el conjunto de imágenes, ya que las mismas aprenden durante su entrenamiento. Debido a su extracción automatizada de características hace que todos los modelos aprendidos sean de una manera más precisa en cuanto a proyectos con visión artificial (Uribe, 2018). Como se muestra en la Figura 17, se tiene la Arquitectura de la red neuronal convolucional, en la que se visualiza el proceso de cada capa, teniendo así, la capa de entrada, en donde se tiene la imagen y se extrae un kernel para formar una matriz de la capa de convolución, seguido de esto se tiene la capa pooling en donde se va tomando de igual manera una porción de la anterior capa; dependiendo de las capas de convolución que se tenga el proceso anterior se va repitiendo. Finalizado este proceso se pasa a las capas de fully connected, teniendo como resultando una función de softmax para tener la capa de salida.

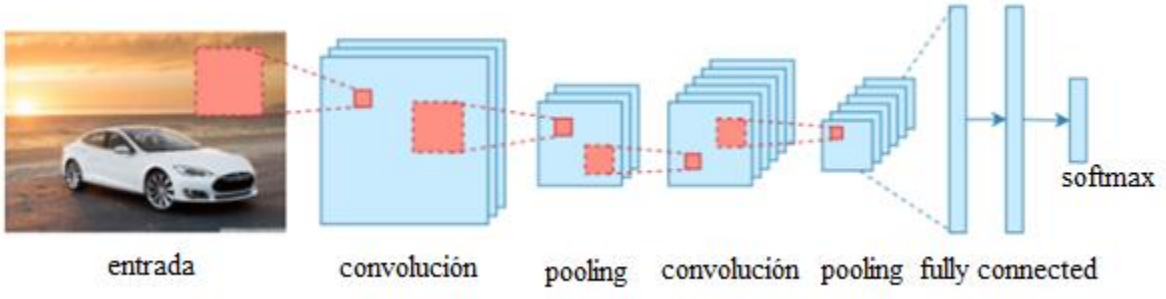

Figura 17. Arquitectura redes neuronales convolucional

Fuente: (Bootcamp AI, 2019); Editado por el autor

Las redes neuronales convolucionales son un algoritmo de *Deep Learning* el cual está diseñado para trabajar con imágenes, las mismas que serán tomadas como entradas, y se les

asignará un peso a ciertos elementos en la imagen; de esta manera se diferenciarán unos de otros. Estas redes contienen varias capas ocultas, donde (Bootcamp AI, 2019) sostiene que, las primeras capas puedan detectar líneas, curvas y así se van especializando hasta llegar a reconocer formas complejas tales como: un rostro, siluetas, etc..

Por otro lado, (Quintero, 2018) enfatiza que las redes neuronales convolucionales están compuestas por múltiples capas. Al inicio se tiene la fase de extracción de características, la cual está compuesta de neuronas convolucionales y de reducción. Conforme se avanza en la red, se disminuyen las dimensiones; de esta manera se activan las características cada vez más complejas. Y al final se encuentran neuronas muchas más sencillas para realizar la clasificación.

La red toma los pixeles[2] de una imagen como entrada; si se tiene una imagen en escala de grises, se toma en cuenta el alto por el ancho, y esto equivaldría al número de neuronas. Ahora, si la imagen es a color, se necesita 3 canales (*red, green, blue*), y se toma en cuenta el alto por el ancho por el número de canales, y esto equivaldría al número de neuronas.

2.5.2.2.1. Operaciones principales de las redes convolucionales

Existen cuatro operaciones principales que conforman una red convolucional; la capa convolucional, rectificador lineal de unidad, sub-muestreo *(subsampling) o pooling* y la capa "totalmente conectada" o fully connected.

- *Capas convolucionales*

La capa convolucional es la encargada de distinguir las redes neuronales convolucionales de cualquier otra red neuronal, dispone de una aplicación para realizar la operación de convolución.

[2] Un pixel es la unidad básica de una imagen digitalizada en pantalla a base de puntos de color o en escala de grises.

Cabe señalar que la convolución consiste en aplicar un kernel o filtro a la imagen que se ingresa, de esta manera se tiene un mapa de todas las características de la imagen original. El kernel se desplaza a través de la imagen de entrada, de tal manera que se deslizará de izquierda a derecha y de arriba hacia abajo, hasta que haya pasado por toda la imagen. De esta manera se calcula el producto entre cada uno de los elementos de la imagen y este elemento del filtro se superpone en esa posición, sumándose todos los resultados (García, 2020).

La Figura 19 muestra los pasos de convolución que se realiza a una imagen de entrada, el proceso de convolución es de la siguiente manera: En primer lugar, el filtro cubre toda la matriz de la imagen en la parte superior izquierda. Seguido de esto, se tiene el producto con relación al elemento del filtro y se toma una sección de la imagen que se encuentra en el filtro; es decir, se tiene el elemento superior izquierdo del filtro, el cual se multiplica por el elemento superior izquierdo de la imagen, este proceso se lo realiza sucesivamente. Luego de esto, se añaden los resultados para obtener un valor. Después el filtro se mueve a lo largo de la imagen, el cual recorre una distancia, y el proceso se repite. Finalmente se tiene una nueva matriz, con diferentes dimensiones. Cabe recalcar que este ejemplo se tiene una matriz de 3 x 3, las dimensiones pueden ser de acuerdo a la configuración que se realice, de igual manera los valores del filtro son los que se configure dentro del proceso. De esta manera se tiene: la capa de partida, el filtro utilizado y finalmente se tiene la capa ya convolucionada.

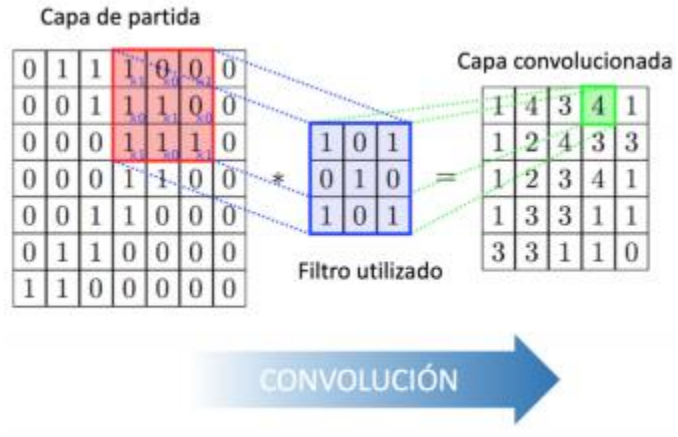

Figura 18. Ejemplo de aplicación de un filtro convolucional

Fuente: (García, 2020)

- Rectificador lineal de unidad
- *El rectificador lineal, es una función de activación, la cual se aplica a todos los elementos del tensor de entrada; convierte en 0 todos los elementos negativos del tensor.* Capa de *pooling*

En la capa de *pooling* se reducen los tamaños de los de mapas de características mediante el uso de funciones. Para ello existen tres maneras de aplicar pooling:

- Max-pooling: devuelve el máximo valor entre todos los píxeles cubiertos por el kernel. En la Figura 20, se aprecia un ejemplo en el que se tiene una matriz con el valor máximo del kernel. Es decir, en el ejemplo se tiene una matriz de 3 x 3, en donde se escoge el valor máximo de dicha matriz, se repite el proceso mediante el desplazamiento del mismo para obtener la matriz de max pooling..
- Min-pooling: devuelve el valor mínimo entre todos los píxeles que se encuentran cubiertos por el kernel.

- Average pooling: Se calcula un valor medio entre todos los píxeles que se encuentran cubiertos por el kernel.

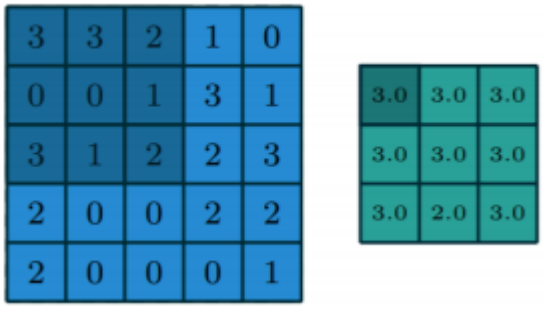

Figura 19. Ejemplo de aplicación de max-pooling

Fuente: (García, 2020)

- Capa clasificadora

Una vez aplicada las capas de convolución, sucesiva y de pooling; las redes utilizan capas completamente conectadas para realizar la inferencia[3]. Y la última capa de la fully connected tendrá tantas neuronas como el número de clases que se quiere predecir.

En la Figura 21 se observa paso a paso las capas de red neuronal convolucional para tener las características necesarias para el respectivo entrenamiento.

[3] Proceso por medio del cual se establecen conclusiones basadas en argumentos corroborables.

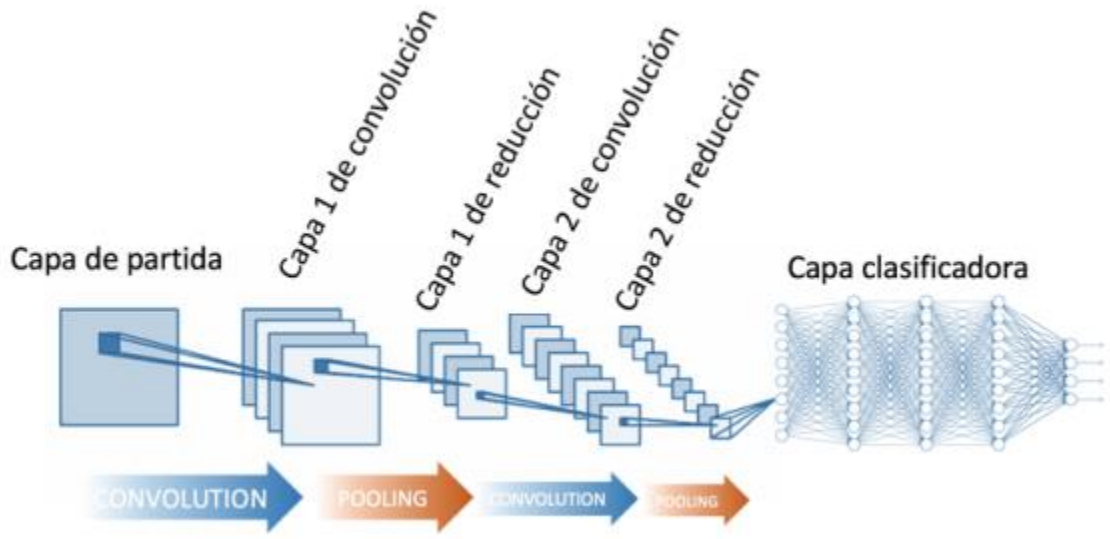

Figura 20. Capas red neuronal convolucional

Fuente: (García, 2020)

La red posee varios parámetros de diseños que se pueden elegir para implementar una CNN, como, por ejemplo, la cantidad de capas convolucionales y *subsampling*[4], como también la cantidad de capas ocultas en la etapa de clasificación. Otro parámetro para elegir también es la función de activación de cada capa. En la Tabla 1, se muestra las Funciones de activación recomendadas para una CNN; en la que se indica los tipos de capas que se tiene y la función de activación que realiza cada una.

[4] Técnica con la cual se comprime la información de color que contiene una señal para favorecer a la información contenida en la luminancia. De esta forma, el ancho de banda queda reducido, pero sin perjudicar la calidad de esta imagen comprimida.

Tabla 1 Funciones de activación para CNN

Tipo de Capa	Función de Activación
Convolucional	Identidad
Max-pooling	Rectificado lineal (Rectified linear)
Fully connected (capa oculta)	Rectificado lineal (Rectified linear)
Fully connected (capa de salida)	Tangente hiperbólica Softmax[5]

Fuente: (Jhan & Restrepo Arteaga, 2015)

[5] Es utilizada como capa final de los clasificadores de múltiples tipos de datos basados en redes neuronales.

CAPÍTULO III. DESARROLLO EXPERIMENTAL.

En este capítulo su objetivo principal es, explicar la elección tanto de Hardware cómo de Software necesarios para el desarrollo del sistema; basándose en el estándar IEEE 29148. Para ello se utilizará la metodología basada en el modelo en V.

3.1. Metodología

Para la realización del proyecto, es necesario realizar una serie de procedimientos, el modelo a utilizarse en el proyecto es el "Modelo en V", dado que satisface la metodología adecuada para la realización del proyecto. Cabe recalcar que esta metodología se basa en el estándar IEEE 29148 el cual posee procesos que permiten determinar los requerimientos tanto de Hardware como de Software.

3.1.1. Modelo en V

(Vallejo, 2016) menciona que, el modelo en V se encuentra realizado para mostrar las fases de desarrollo y pruebas para establecer requerimientos de un sistema propuesto; por otro lado (Pérez, 2012) describe que el modelo en V es una representación gráfica del ciclo de vida para el desarrollo del sistema; es decir resume los pasos principales que se toma junto a los sistemas de validación. En la Figura 22 se aprecia las etapas de desarrollo del modelo en V, mencionadas a continuación:

- En la parte izquierda de la V se definen las especificaciones del sistema.
- En la parte derecha de la V se comprueba el sistema en relación con las especificaciones establecidas en la parte izquierda.
- En la parte de abajo, donde se encuentran las dos partes, es decir representa el desarrollo.

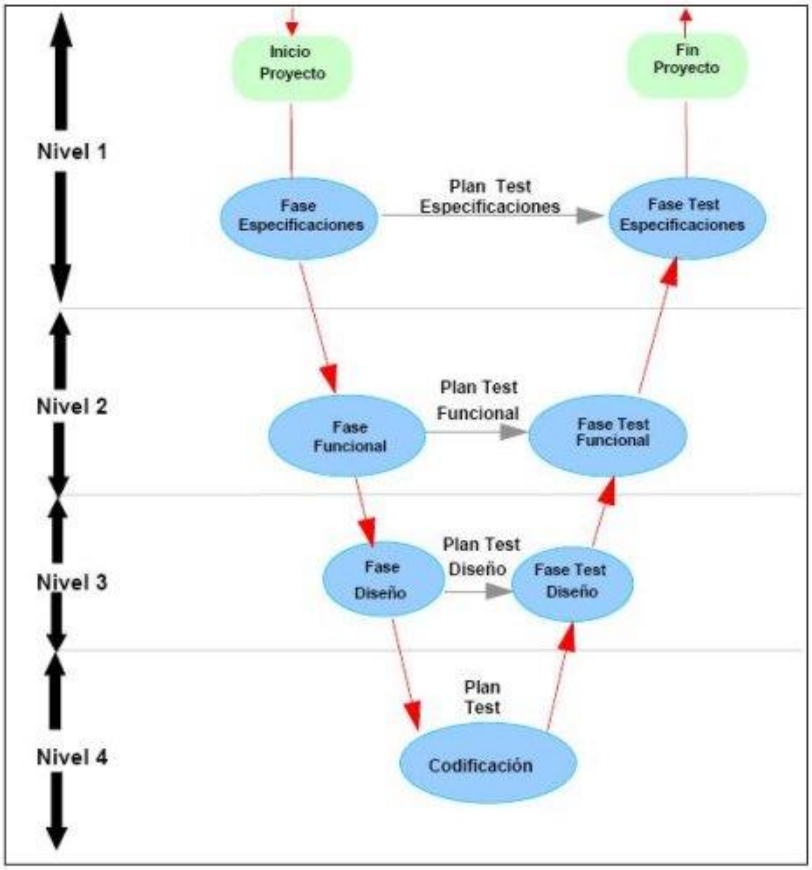

Figura 21. Etapas del modelo en V

Fuente: (Pérez, 2012)

En los 4 niveles lógicos, para cada fase del desarrollo se tiene una fase que corresponde a la verificación. Se establece que dicha estructura, desde el principio y para cada una de las fases del desarrollo existe un resultado verificable. A continuación, se detalla cada nivel del modelo en V.

- NIVEL 1: este nivel se orienta al cliente; tanto el inicio del proyecto como el fin del proyecto. El nivel 1 está compuesto de un análisis tanto de requisitos como de especificaciones.

- NIVEL 2: en este nivel se establecen las características funcionales del sistema que se proponen, ya que se establecen funciones directa o indirectamente, es decir, se vuelve en un documento de análisis funcional.
- NIVEL 3: en el nivel 3 se definen los componentes tanto de hardware como de software del sistema final, más conocido como arquitectura del sistema.
- NIVEL 4: el último nivel es la fase de implementación, en el que se desarrollan los módulos del programa.

(Godoy, 2016), considera que, una vez mencionados los niveles de la metodología del modelo en V; también se debe conocer los procesos que se tiene en cada fase para la validación del sistema, los mismos que se mencionan a continuación: Análisis, Requerimientos, Diseño, Programación, Integración, Verificación e Implementación.

3.2. Análisis.

En esta fase se establecen los requerimientos a considerar en el sistema, basándose en que permitan la determinación de los requisitos tanto del Hardware como del Software. Con el análisis y la investigación que se realiza se dará solución a la problemática planteada;

3.2.1. Situación Actual.

En la actualidad, el reto que tiene el sector agrícola, es la necesidad por desarrollar innovaciones tecnológicas que aporten y favorezcan el desarrollo y la competitividad en dicho sector (Gómez, 2015).

Según los datos estadísticos, (Magap, 2017) indica que: de una producción plantada de 1970 ha, existe una superficie cosechada de 1954 ha, teniendo 16 ha no cosechadas, mientras que (Andrea et al., 2018) menciona que, las pérdidas que existen en los cultivos de tomate riñón, debido

a la presencia de maleza, se ve afectada, ya que la agricultura en el Ecuador tiene falta de recursos tecnológicos y aplicaciones que hacen que sea imposible explotar su verdadero potencial como productor agrícola.

Basados en los datos expuestos, se ve la necesidad de disponer de procesos que innoven el área de la agricultura, en este caso, reconocer malezas dentro de un cultivo de tomate riñón. En la actualidad, utilizando la tecnología, ayuda a tener mejores resultados, de esta manera tener cultivos con mayor producción(Gómez, 2015).

Para realizar el sistema propuesto, se ha considerado los cultivos de tomate riñón de invernadero mencionados a continuación: Cultivo de tomate riñón de invernadero "TOLAS DE SOCAPAMBA" vía Yahuarcocha, Cultivo de tomate riñón "EL ROSAL" ubicado en Pimampiro y cultivo de tomate riñón "CHALTURA"; los mismos que favorecieron para la toma de datos y pruebas respectivamente.

3.2.2. Dimensionamiento de Stakeholders

En primer lugar, se dimensionan los stakeholders implicados, directa o indirectamente que se necesitan para el desarrollo del sistema; sus especificaciones se muestran en la Tabla 2.

Tabla 2. Stakeholders

Stakeholders	
Usuarios directos	UTN
Usuarios indirectos	Sr. Wilson Suárez
	Sr. Karlos López
Administradores	Sr. Calisto Miranda
	Msc. Edgar Maya
Director y Fiscalizadores	Msc. Mauricio Domínguez
	Msc. Luis Suárez
Desarrollador	Srta. Karla Moncayo

Fuente: Autoría propia

3.2.3. Técnicas de recolección de Información

Las técnicas de recolección de información, son todas las actividades y procedimientos que brindan el acceso directo a la persona investigadora para tener acceso a toda la información necesaria y con ello cumpla con los objetivos necesarios investigativos para los requerimientos a usar (Terán, 2020)

En las técnicas de recolección de información se mencionan las siguientes:

- **Observación:** En esta técnica se extrae de forma metódica y a través de la revisión, los datos.
- **Recopilación documental:** En esta técnica se obtiene datos e información a partir de fuentes documentales con el fin de ser utilizados.
- **Entrevista:** Es una técnica para recabar datos; se define como una conversación que se propone un fin determinado distinto al simple hecho de conversar. Es un instrumento técnico que adopte la forma de un diálogo.

- **Encuesta:** Es una técnica de recopilación de información donde el investigador interroga a los investigados los datos que desea obtener ya sea mediante cuestionarios, preguntas, etc.

Para este proyecto se emplean las técnicas de observación, la entrevista y la encuesta; las cuales servirán de herramientas para tener información sobre el sistema a realizarse.

3.2.3.1. Observación.

Mediante la técnica de observación, se define el tipo de maleza con la que se va a realizar el sistema de entrenamiento, para su respectivo reconocimiento. En este caso teóricamente se define algunos tipos de malezas que existen en el Ecuador, para ello se vistita tres cultivos de tomate riñón y se verifica que tipo de malezas existe en cada uno de ellos con la ayuda de los agricultores; una vez recopilada la información se elige una maleza en específico llamada "Ipomoea" o más conocida como Campanilla debido a sus características, ya que por su forma y textura se la puede diferenciar de la planta y su entrenamiento en el sistema se vuelve eficiente. Cabe recalcar que el hábitat de este tipo de malezas es en zonas tropicales, por lo que la adquisición de datos se lostoma del cultivo ubicado en Chaltura.

3.2.3.2. Entrevista y encuesta.

Para determinar el objetivo del proyecto a realizarse, se utiliza la encuesta, la cual nos permite determinar y evaluar las necesidades dentro de un cultivo de tomate riñón. Para la entrevista se utilizó preguntas en específico, como:

- ¿Cada que Tiempo siembran el tomate riñón?
- ¿Cuánto tiempo tarda el tomate, para ser cosechado?
- ¿Qué tipos de maleza que existen en el cultivo?

- ¿Cada que tiempo crecen las malezas?
- ¿Cada que tiempo se podan las malezas?
- ¿Qué daños ocasiona la presencia de malezas en los cultivos de tomate riñón?

Una vez realizadas las preguntas anteriormente mencionadas; se tiene el resultado de la entrevista Anexo1, realizada al Sr. Wilson Suárez agricultor y encargado de los cultivos de tomate riñón en "TOLAS DE SOCAPAMBA" y al Sr. Calisto Miranda agricultor y encargado del cultivo de tomate riñón en Chaltura. De igual manera se plantea una encuesta que permita brindar información necesaria para el diseño del sistema. La encuesta es realizada y dirigida a los agricultores en cultivos de tomate riñón de invernadero; en este caso se realiza preguntas cerradas de opción múltiple, las mismas que se encuentra detalladas en el Anexo 2 y los resultados de la encuesta realizada se encuentra en el Anexo 3.

3.3. Requerimientos.

En los requerimientos se maneja el estándar IEEE29148, estándar que permite crear las necesidades de los stakeholders, convirtiéndolos en requisitos que sirven para el diseño y la funcionalidad del sistema, utilizando procesos iterativos y recursivos a lo largo del ciclo de vida del proyecto. El estándar IEEE29148 contiene disposiciones para todos los procesos relacionados con la ingeniería de requisitos para sistemas y productos y servicios de software a lo largo del ciclo de vida (ISO/IEC/IEEE 29148, 2011).

3.3.1. Nomenclatura de requerimientos.

A continuación, la Tabla 3 detalla las abreviaturas utilizadas para una mejor comprensión y administración de la información acorde a los requerimientos usados para el sistema.

Tabla 3. Abreviatura de los requerimientos

Abreviatura	Requerimiento
StSR	Requerimiento de Stakeholders
SySR	Requerimientos del Sistema
SRSH	Requerimientos de Arquitectura

3.3.2. Requerimientos de los Stakeholders.

En la Tabla 4 se detallan los requerimientos de los stakeholders para la realización del sistema.

Tabla 4. Requerimientos Operacionales y de Usuarios

#	Requerimientos Operacionales	Prioridad Alta	Media	Baja
	Requerimientos de Stakeholders (StSR)			
StSR1	Disponibilidad de un punto de energía eléctrica, para la alimentación del raspberry pi 4.	X		
StSR2	Disponibilidad de una carpeta dentro del dispositivo para albergar las imágenes reconocidas	X		
StSR3	Ubicar el raspberry pi 4, para la realización de pruebas a escala de laboratorio.	X		
	Requerimientos de Usuarios			
StSR4	La información del reconocimiento de la maleza Ipomoea debe mostrarse en un formato que comprenda el usuario	X		
StSR5	Visualizar las imágenes reconocidas dentro del Bounding box	X		
StSR6	El dispositivo debe ubicarse de tal manera que el sistema pueda realizar el reconocimiento de las mismas a escala de laboratorio.	X		

3.3.3. Requerimientos del sistema.

Tabla 5. Requerimientos de Uso, Interfaces, Estados, Físicos

Requerimientos del sistema (SySR)				
#		Prioridad		
	Requerimientos de uso	Alta	Media	Baja
SySR1	El dispositivo debe estar encendido para llevar a cabo el proceso de reconocimiento	X		
SySR2	El algoritmo debe ser ejecutado para dar inicio al reconocimiento	X		
SySR3	El sistema debe tener la respuesta del reconocimiento, mediante el autoguardado dentro de la carpeta en el dispositivo	X		
Requerimientos de Interfaces				
SySR4	Disponer de conexión a cámara	X		
SySR5	Debe incluir puertos de E/S para la conexión correcta de los periféricos	X		
Requerimientos de Estados				
SySR6	Debe estar encendido el Raspberry pi 4 para su uso	X		
SySR7	La cámara debe estar encendida para realizar el reconocimiento de la maleza Ipomoea.	X		
SySR8	El sistema debe ejecutarse para realizar el reconocimiento respectivo	X		
Requerimientos Físicos				
SySR9	El dispositivo debe ser portable a un lugar donde se disponga de corriente eléctrica para su funcionamiento	X		
SySR10	Se debe contar con el lugar para realizar el reconocimiento de la maleza Ipomoea.	X		
SySR11	Realizar las pruebas a escala de laboratorio	X		

3.3.4. Requerimientos de arquitectura.

Tabla 6. Requerimientos Lógicos, Software, Hardware, y Eléctricos.

Requerimientos de Arquitectura (SRSH)				
#		Prioridad		
	Requerimientos Lógicos	Alta	Media	Baja
SRSH1	Debe existir la compatibilidad tanto del Software para el funcionamiento correcto	X		
SRSH2	Debe existir la compatibilidad tanto del Hardware para el funcionamiento correcto	X		
	Requerimientos de Hardware			
SRSH3	Se debe contar con una pantalla con entrada a HDMI para su conexión directa con el raspberry	X		
SRSH4	Disponer de conexión de cámara en el raspberry	X		
SRSH5	Conexión eléctrica tanto para la pantalla como para el raspberry	X		
SRSH6	Capaz de procesar datos de alta velocidad	X		
SRSH7	Usar un procesador, capaz de reconocer las imágenes en la realización del sistema.	X		
	Requerimientos de Software			
SRSH8	Debe ser compatible con GNU/Linux	X		
SRSH9	Debe soportar librerías extensas	X		
SRSH10	Debe ser óptimo en el consumo de la RAM, para no sobrecargar el sistema embebido	X		
SRSH11	Soporte en tiempo real	X		
SRSH12	Debe ser gratuito	X		
	Requerimientos Eléctricos			
SRSH13	Disponer de tomacorriente para la conexión de la pantalla a utilizarse	X		
SRSH14	Disponer de tomacorriente para la conexión del raspberry a utilizarse	X		

3.3.5. Propósito y ámbito del sistema.

Se propone el desarrollo de un sistema de visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero; en este caso se ha tomado un tipo de maleza en específico "*Ipomoea purpurea"*. Esta maleza tiene características que se distinguen de la planta de tomate riñón, por su forma y tamaño. El sistema basado en el dataset creado a partir de dicha maleza, resulta eficiente al momento del entrenamiento, ya que, en el etiquetado se guarda las características deseadas para su reconocimiento. Una vez realizado el entrenamiento respectivo del sistema, se procede a realizar las pruebas a escala de laboratorio.

3.3.6. Descripción general del sistema.

El sistema de visión por computador está compuesto por una pantalla donde se visualiza los resultados del reconocimiento de la maleza Ipomoea; dicha pantalla se encuentra conectada al dispositivo Raspberry Pi 4, en donde se realiza el entrenamiento mediante redes neuronales previo al reconocimiento final que realiza el sistema.

Para la realización del sistema, se adquiere fotografías de la maleza Ipomoea en el cultivo de tomate riñón de invernadero, para el entrenamiento respectivo. En las pruebas del sistema, a escala de laboratorio se dispone de una conexión directa de la cámara adquirida hacia el raspberry 4 con el objetivo de tener el sistema de reconocimiento, mediante la toma de fotografías guardadas, videos guardados o a su vez en tiempo real; el resultado del reconocimiento se observará en una carpeta albergada en el mismo sistema.

3.4. Elección del Software.

Se basa en los requisitos de Software establecidos anteriormente y se procede a elegir el Software necesario para la realización del proyecto.

3.4.1. Lenguaje de programación.

Una vez detallado los requerimientos para la elección del lenguaje de programación, se describe cada requerimiento y se determina la opción adecuada para el desarrollo del proyecto. Todas las características de los lenguajes de programación se detallan en el Anexo 04; los cuales servirán para analizar los requerimientos.

Para dar paso a la selección del lenguaje de programación se presenta un análisis en donde se toma en cuenta los requerimientos de software en la Tabla 7.

Tabla 7. Tabla Comparativa de los lenguajes de programación, basados en los requerimientos.

Requerimientos	Python	Matlab
SRSH8	1	0
SRSH9	1	1
SRSH10	1	1
SRSH11	1	0
SRSH12	1	0
Valoración	5	2
Cumplimiento: 0 – No cumple 1 – Si cumple		

Fuente: Autoría propia

Del análisis realizado en la Tabla 7 y una vez mencionadas las características en el Anexo 04. para el presente trabajo se escoge Python; ya que da una calificación óptima, debido a algunas de sus características como: su soporte es en tiempo real, al igual que es un Software libre, dispone

de extensas y potentes librerías, A diferencia de Matlab que se utiliza para proyectos robustos, pero la librería estándar no es completa para la utilización en modo free.

3.4.2. Algoritmo.

Por otro lado, se necesita de un algoritmo para clasificar las imágenes y detectar el objeto deseado; es decir, el algoritmo lo que hace es, añadir la parte de la detección (cuadro con las posiciones x e y, alto y ancho del objeto encontrado).

Para lo cual se desarrolla una comparativa en la Tabla 8, basados en las características mencionadas en el Anexo 5; los cuales servirán para analizar los requerimientos.

Tabla 8. Tabla Comparativa de los algoritmos, basados en los requerimientos.

Requerimientos	YoloV3	YoloV3 Tiny
SRSH8	1	1
SRSH9	1	1
SRSH10	0	1
SRSH11	1	1
SRSH12	1	1
Valoración	4	5
Cumplimiento:		
0 – No cumple		
1 – Si cumple		

Fuente: Autoría propia

Del análisis comparativo se escoge YoloV3 Tiny Python, ya que tiene características como: menos iteraciones para obtener una alta precisión en la detección, velocidad de inferencia [6]que

[6] Proceso interpretativo efectuado para deducir el significado implícito de los datos obtenidos de un contexto en específico.

hace que supere la precisión sobre los datos, además es, hasta cinco veces más rápida que YOLOv3, por lo que hace que sea mejor que Yolov3.

3.5. Elección del Hardware.

A continuación, en base a los requerimientos de Stakeholders, requerimientos de sistema y arquitectura antes mencionados, se realiza la selección de las mismas; tomando en cuenta sus especificaciones y costos de manera detallada.

3.5.1. Benchmarking.

Para realizar esta comparación se toma en cuenta las características técnicas de los componentes de Hardware para realizar su respectivo análisis. Se detalla la información utilizada en el Benchmarking en el Anexo 6.

3.5.1.1. Sistema Embebido.

A continuación, se realiza un benchmarking, lo cual consiste en una comparación de características técnicas de los diferentes hardware que impacten, entre las principales se mencionan en la Tabla 9 como: Raspberry pi Zero W, Raspberry 3B+ y Raspberry Pi 4

Tabla 9. Tabla de características entre Raspberry Pi Zero, Raspberry pi 3 B+ y Raspberry Pi 4.

Placa	**Raspberry Zero W**	**Raspberry 3 B+**	**Raspberry Pi 4**
Parámetros	Características	Características	Características
Costo	$45	54	$89.99
Procesador	BCM2835 ARM1176JZF-S single core 1Ghz	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC	ARM Cortex-A72
Memoria RAM	512 Mb	1GB LPDDR2 SDRAM	1 GB / 2 GB / 4 GB LPDDR4 SDRAM
WLAN	Wi-Fi 802.11b / g / n	Wifi	Wi-Fi 802.11ac
Bluetooth	Bluetooth 4.1 BLE	Bluetooth 4.2, BLE	Bluetooth 5.0
LAN	No dispone	Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)	Si dispone
Puertos	2x micro-USB ports (one for power)	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Tomas auriculares / vídeo compuesto Micro SD Micro USB (alimentación) Power-over-Ethernet (PoE)	2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Micro SD Conector de audio jack USB-C (alimentación)
Acceso	Extended 40-pin GPIO header	-	GPIO 40 pines
Interfaces para video	Mini HDMI port; CSI camera port; composite video port.	-	Micro HDMI, USB 2.0 , USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Micro SD Conector de audio jack USB-C (alimentación)
Dimensiones	65 x 30 x 5 (mm)		85.6 x 56.5 x 17 (mm)
Consumo	5v/200 - 350 mA		5v/300mA - 1.3A

Fuente: Autoría propia

Una vez realizada la tabla de características entre las placas electrónicas mencionadas anteriormente, se puede notar que el Raspberry Pi 4, dispone de un procesador con mejores

características que el modelo de Raspberry pi Zero y el Raspberry Pi 3 B+, ofreciendo mejor desempeño para realizar las tareas para el desarrollo del proyectoen mención, por otro lado, el Raspberry Pi 4 dispone de mayor capacidad de Memoria RAM mejorando su rendimiento en comparación a los otros dos dispositivos. Por otro lado, se puede mencionar que el Raspberry Pi Zero, dispone de un tamaño reducido a comparación del modelo Raspberry Pi 3+ y el Raspberry Pi 4 lo que hace imposible disponer de características como las que se mencionan de los otros dispositivos. Es por ello que se realiza la comparativa en la Tabla 10.

- Selección de la placa

Tabla 10. Tabla de Elección de placa.

Requerimientos						
Hardware de desarrollo	SRSH3	SRSH4	SRSH5	SRSH6	SySR7	Valoración
Raspberry Pi Zero W	1	1	1	0	1	4
Raspberry Pi 3 B+	1	1	1	1	0	4
Raspberry Pi 4	1	1	1	1	1	5
Cumplimiento: 1-Cumple 0-No cumple						

Fuente: Autoría propia

Una vez realizada la tabla comparativa, se concluye que el modelo Rasberry Pi 4 cumple con mejor satisfacción los requisitos a comparación del modelo Pi Zero y el Raspberry pi 3+, los cuales no poseen un procesamiento suficiente y necesario para la realización del proyecto, por lo

que bajaría el procesamiento y no podría funcionar el sistema con la eficiencia adecuada, siendo el principal motivo para descartarlos ya que la capacidad de procesamiento del modelo Pi 4 posee una arquitectura más robusta que el modelo Pi Zero.

3.6. Diseño del sistema.

Una vez que se definen los requerimientos de Hardware y Software, se procede a realizar el diseño del sistema; tomando en cuenta cada uno de los criterios mencionados anteriormente; los mismos que permitirán el desarrollo e implementación del sistema de reconocimiento de maleza de tomate riñón

En esta etapa se da a conocer mediante diagramas las funciones del sistema; los cuales servirán de guía para realizar ordenadamente los procesos a seguir.

3.6.1. Diagrama de bloques del sistema

En este apartado se realiza el diagrama de bloques del sistema de reconocimiento de la maleza *Ipomoea*. En el diagrama se tiene dos etapas, en las cuales consta el proceso para la realización del sistema y la ejecución del mismo. En el proceso de la realización del sistema, se establece la secuencia de pasos, iniciando con la adquisición de imágenes a partir de datos tomados de la maleza en mención del cultivo de tomate riñón. A continuación, se realiza el etiquetamiento de la maleza *Ipomoea*, mediante el Software libre *LabelImg*; y para el entrenamiento del sistema se lo realiza mediante, la arquitectura CNN. En la ejecución del sistema se extraen las características de la maleza *Ipomoea*, ya sea en tiempo real, fotos o videos guardados; en base al previo entrenamiento para su reconocimiento.

En la Figura 23 se observa el diagrama de bloques del sistema de reconocimiento de la maleza *Ipomoea* en cultivos de tomate riñón de invernadero, logrando una mejor comprensión de los procesos mencionados anteriormente.

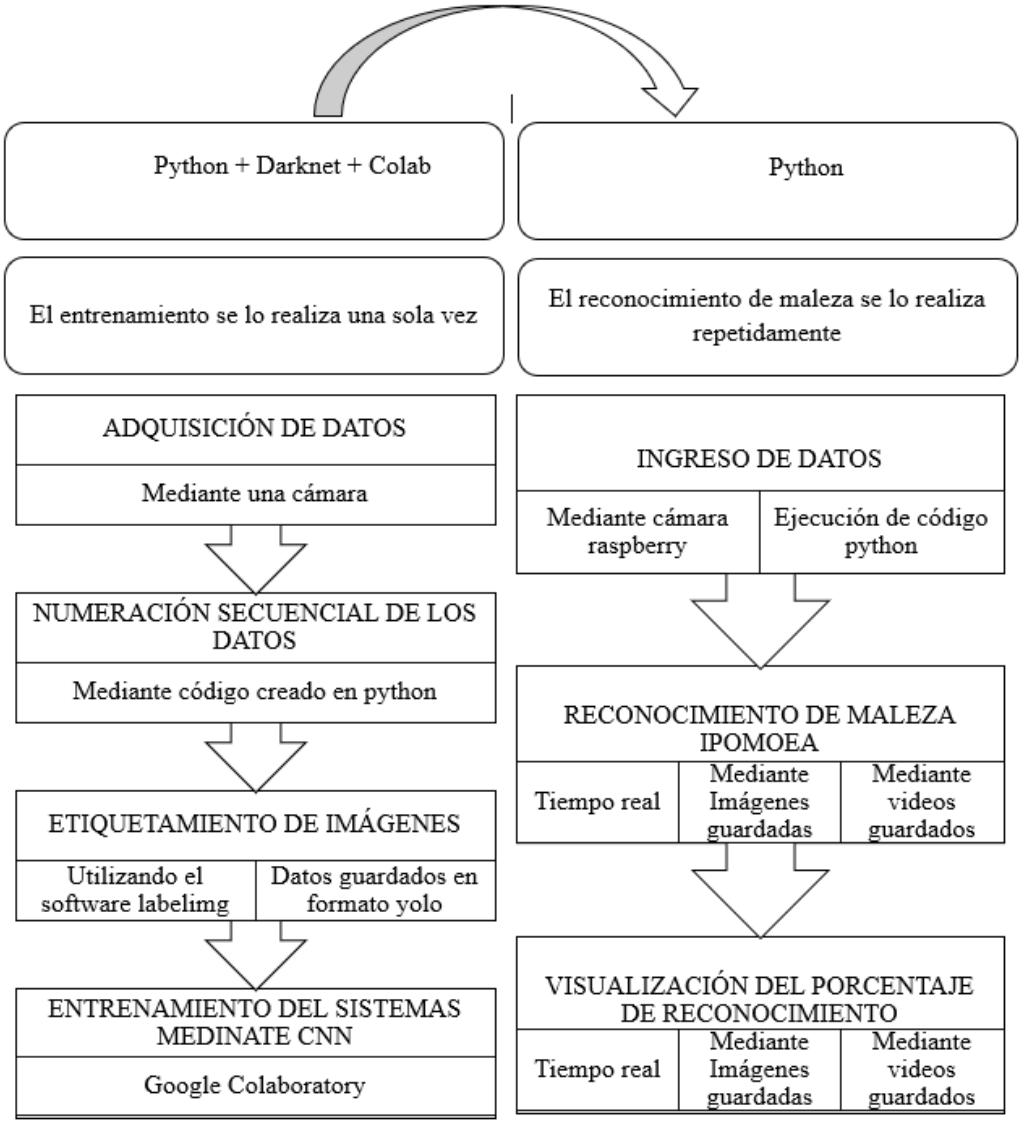

Figura 22. Diagrama de bloques Sistema

Fuente: Autoría propia

3.6.1.1. Diagrama de funcionamiento del sistema.

En el diagrama de funcionamiento se especifica el diseño del Software para la realización del sistema de visión por computador. En esta etapa se especifica el diagrama del sistema, el cual, se ha subdividido en dos partes. El diagrama Nro. 1 se muestra en la figura 24, inicia con la

adquisición de datos, que consiste en la obtención del conjunto de fotografías de la maleza *Ipomoea* en los cultivos de tomate riñón. Seguido de esto, se procede a etiquetar cada una de las imágenes mediante el Software LabelImg (herramienta de anotación de imágenes gráficas); el cual proporciona imágenes con cuadros delimitadores después del etiquetado, los datos etiquetados de salida se los guarda en formato Yolo para su compatibilidad en la realización del sistema.

A continuación, se realiza el entrenamiento del sistema mediante redes neuronales, con la utilización del lenguaje de programación de software libre Python; en este caso se utiliza la herramienta Google Colab ya que proporciona GPU gratuito, logrando tener un procesamiento efectivo para el entrenamiento adecuado del sistema.

PROCESS OFFLINE

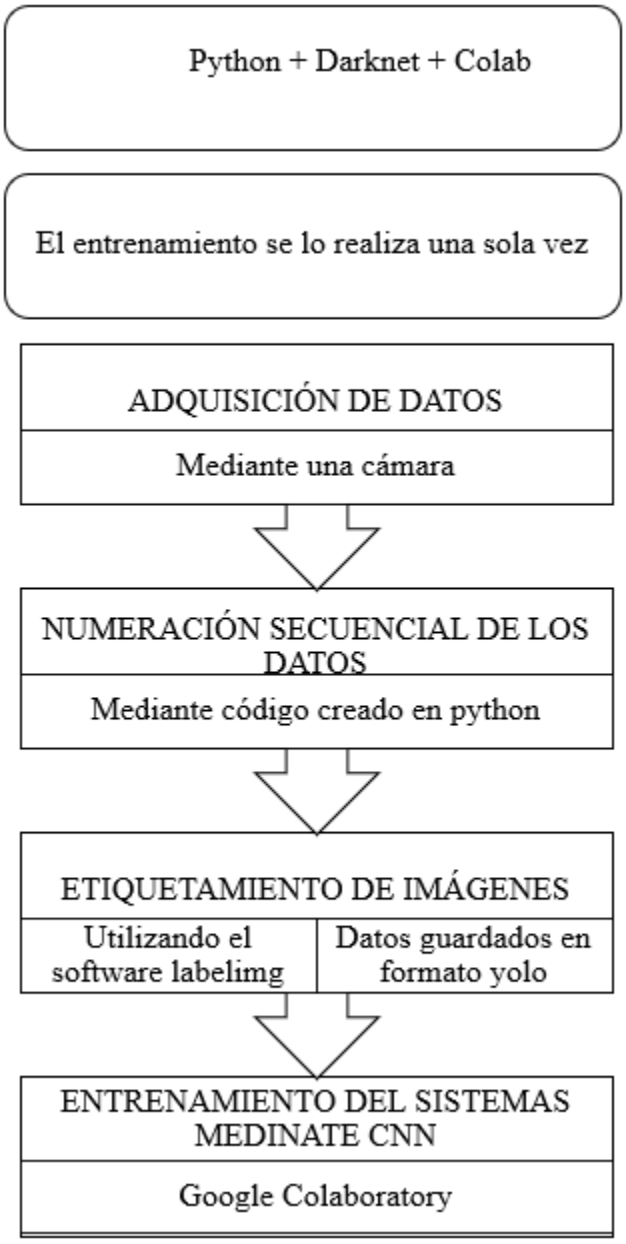

Figura 23. Diagrama Nro. 1 del Funcionamiento del sistema

Fuente: Autoría propia

Para el diagrama No2 se tiene la cámara conectada directamente al raspberry Pi 4; la misma que toma la foto e ingresa en pixeles a la capa de entrada de la red neuronal. Una vez que ingresan todos los pixeles en tres dimensiones(rojo-verde-azul), se da paso a la capa convolucional, la cual se encarga de detectar las características de la imagen, las mismas que pueden ser: textura, tonalidad, forma, línea, entre otros. El proceso se realiza como si fueran filtros, los mismos se encargan de extraer todos los atributos de la imagen, facilitando la clasificación en las capas posteriores.

Seguido de este proceso se realiza el *pooling* (operación que sirve para reducir el tamaño de la imagen), en esta capa se extraen solamente los valores más representativos para optimizar los cálculos de la red neuronal. A continuación, se realiza el *flattening*[7], en donde se cambia el formato para que pueda ser leído por las capas de neuronas posteriores.

Para la capa de neuronas, en donde se incluyen pesos y una función de activación para cada una, se tiene el proceso de clasificación de las imágenes, en donde aprenden a relacionar los atributos de las otras capas con la predicción. Por último, se tiene la capa de salida en donde nos devuelve una probabilidad para cada uno de los valores, en este caso el resultado a obtener sería el reconocimiento de la maleza Ipomoea. Para ello se tiene la Figura 25, donde se presenta el diagrama Nro 2 del funcionamiento del sistema; en el cual se ingresan los datos a ser reconocidos ya sea en tiempo real, mediante imágenes guardadas o un video guardado.

[7] Se realiza para convertir los datos multidimensionales en un vector de características 1D para ser utilizado por la siguiente capa que es la capa densa

PROCESS ONLINE

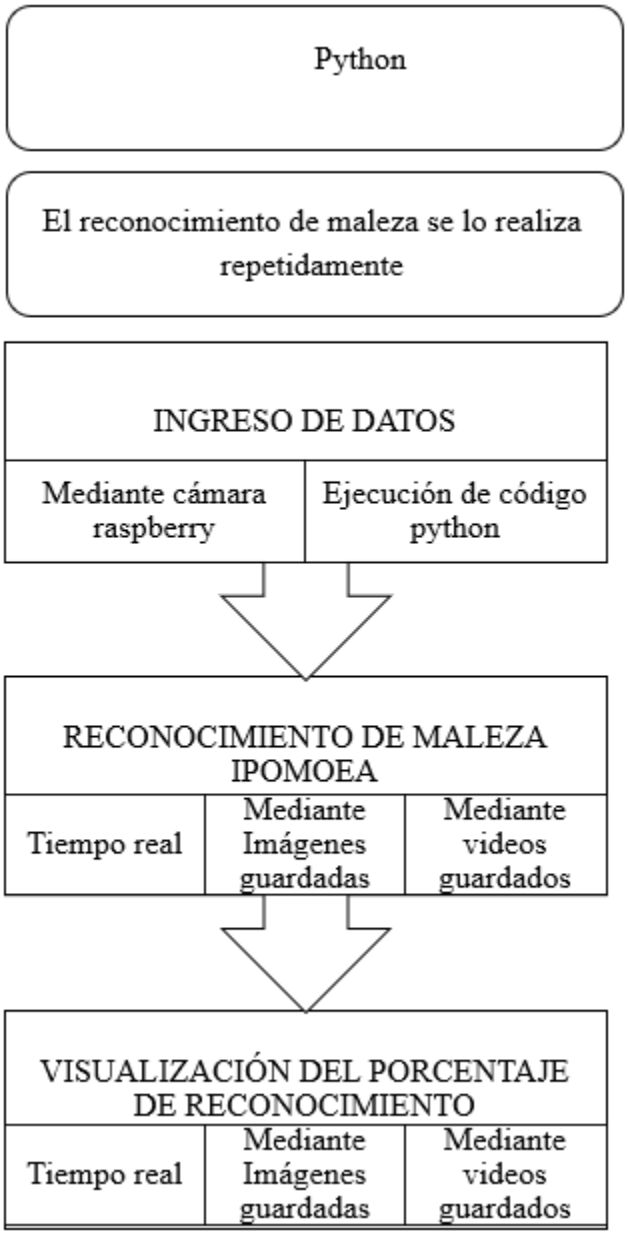

Figura 24. Diagrama Nro 2 del funcionamiento del sistema

Fuente: Autoría propia

3.6.2. Diagrama de flujo del sistema

En este apartado se realiza el diagrama de flujo del sistema de reconocimiento de la maleza *Ipomoea*. En el diagrama de flujo se inicializa con el código creado en Python; una vez que se ingresa a la carpeta en donde se encuentra el código, se define si la prueba a escala de laboratorio se la va a realizar en tiempo real, con imágenes guardadas o con videos guardados. Una vez que se escoja una de las opciones se procede a ejecutar el código deseado en donde se visualizará el porcentaje de reconocimiento que se obtiene en cada maleza Ipomoea encontrada.

En la Figura 26 se muestra el diagrama de flujo del sistema de reconocimiento de la maleza *Ipomoea* en cultivos de tomate riñón de invernadero, logrando una mejor comprensión del proceso mencionado anteriormente.

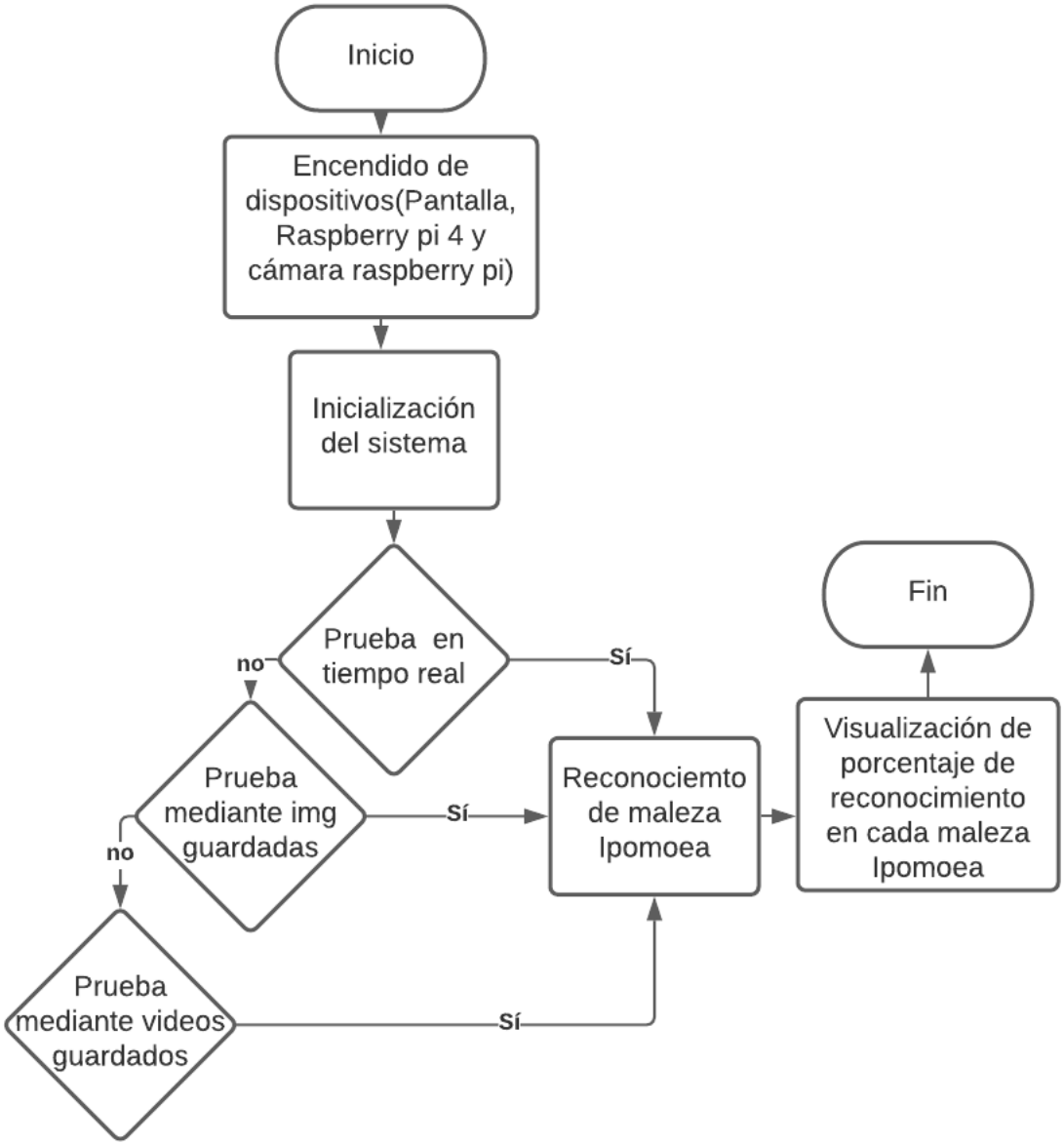

Figura 25. Diagrama de flujo del Sistema

Fuente: Autoría propia

3.6.3. Diagrama de la Arquitectura del sistema

El sistema cuenta con un sistema embebido central que se encuentra conectado directamente a una pantalla, en la que se visualiza la ejecución del mismo. De igual manera, se tiene una cámara raspberry pi conectada directamente al sistema embebido, la cual permite realizar el reconocimiento de la maleza Ipomoea, ya sea en tiempo real, mediante imágenes o videos

guardados. Cabe recalcar que para monitorear y dar inicio al sistema se dispone de un teclado USB conectado en el raspberry Pi 4; al igual que se dispone de una fuente de alimentación para la pantalla como también para el raspberry pi4.

En la Figura 27 se observa el diagrama de la arquitectura del sistema de reconocimiento, detallando el procedimiento que se realiza para obtener los resultados esperados y el correcto funcionamiento.

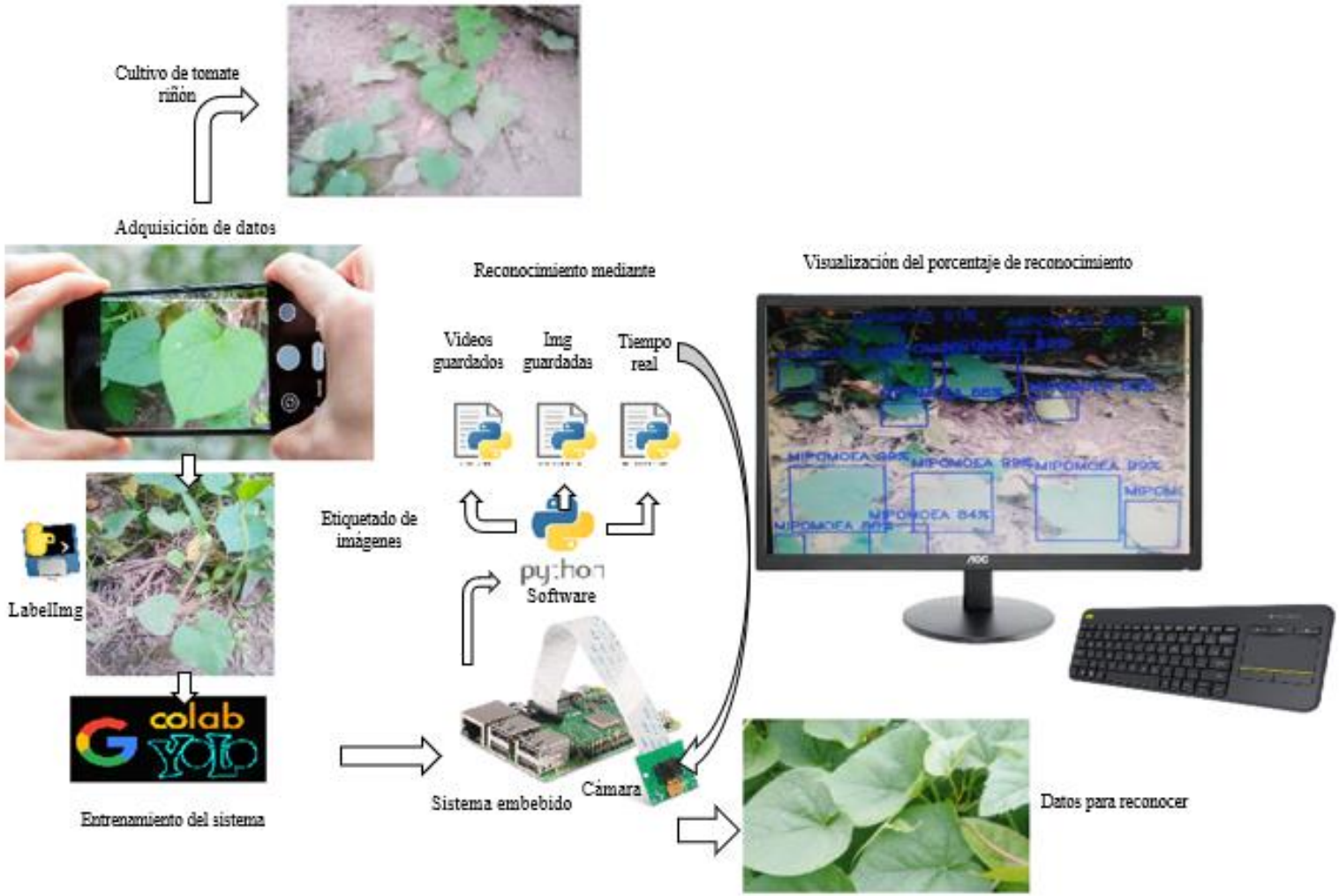

Figura 26. Diagrama de la arquitectura del sistema

Fuente: Autoría propia

3.6.4. Desarrollo del Software

En este apartado se procede a desarrollar cada una de las etapas que tiene el software de programación en todo el procedimiento de visión artificial. Para su desarrollo se inicia con el reconocimiento de la maleza Ipomoea, utilizando el sistema Yolo Tiny V3. Se describe el sistema de reconocimiento de objetos Yolo Tiny V3, la configuración para Darknet, de igual manera se describe la forma de crear los patrones de entrenamiento, entrenar un modelo, y por último, se realiza la prueba del modelo y se reconoce la maleza Ipomoea en tiempo real, mediante imágenes o un video guardado con el modelo entrenado en Yolo Tiny V3.

3.6.4.1. Yolo

Yolo es una sola red convolucional, la cual predice conjuntamente múltiples cuadros delimitadores y varias probabilidades de clase. Se puede mencionar que Yolo utiliza las características de toda la imagen para predecir cada cuadro delimitador; de igual manera predice simultáneamente todos los cuadros delimitadores de cada clase para una imagen. Yolo divide la imagen que se tiene de entrada en una cuadrícula S × S (el tamaño se lo configura acorde a cada criterio); si en la cuadrícula se ubica el centro de la maleza, la celda es responsable de detectar la maleza. Debido a que Yolo se entrena con imágenes completas optimiza el rendimiento de detección. La red de detección tiene capas convolucionales, seguidas de las capas completamente conectadas; al alternar las capas convolucionales se reduce el espacio de características de las capas anteriores.

Para Obtener la versión de Yolo Tiny de Darknet se puede descargar del sitio https://github.com/riclombar/Instalacion_YOLO; en el mismo sitio se encuentran los pesos pre-entrenados.

3.6.4.2. Configuración Darknet de Yolo Tiny V3

Una vez instalado *Darknet*, puede ser ejecutado en algunos sistemas operativos como: Windows, Mac OS y Linux. En este caso el entrenamiento se lo va a realizar en Google *Colaboratory* que permite escribir y ejecutar código de Python directamente desde un navegador, teniendo algunas particularidades como: no se requiere configuración, dispone acceso gratuito a GPU y facilidad para compartir. *Colab* utiliza un entorno interactivo denominado notebook de *Colab*, que permite escribir y ejecutar código. Para este proceso, se utiliza un ordenador con 12 GB de memoria y un procesador Intel Core i7.

3.6.4.3. Adquisición de imágenes

Para realizar el sistema propuesto y ser entrenado mediante Redes Neuronales Convolucionales, en el primer proceso se tiene la adquisición de fotografías realizadas en un cultivo de tomate riñón, considerando la maleza Ipomoea .

En la adquisición de datos, en un inicio se la realiza con un total de 925 imágenes de alta resolución, la cual presenta falencias en el tamaño de la imagen de salida y en el entrenamiento, ya que se utiliza mayor procesamiento. Una vez realizadas las pruebas con las imágenes adquiridas de alta resolución, sin tener resultados favorables se procede a cambiar de tamaño las imágenes entre 921*691 hasta 960 * 720. Considerando que las imágenes son a color, se tiene los datos de entrada de tres canales, los cuales deben ser multiplicados por el número de pixeles. la misma.

Para la toma de datos se realiza un total de 925 imágenes, como se muestra en la Figura 28.

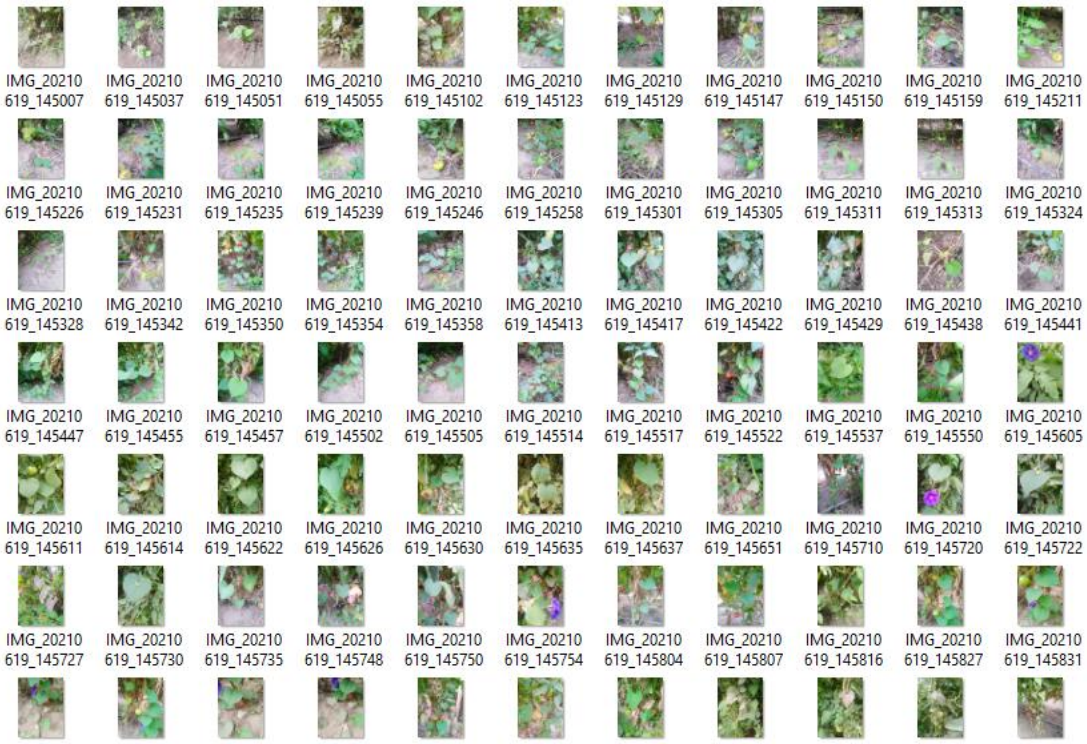

Figura 27. Base de datos de la adquisición de imágenes

Fuente: Autoría propia

Cabe mencionar que, una vez que se tiene la base de datos, se procede a enumerarlas acorde a un nombre secuencial, las cuales se encuentran albergadas en una carpeta con nombre TrainMIpomoea. Para ello se ha tomado en cuenta un código realizado en el Software libre Python, el cual facilita asignarles el nombre a todas las imágenes secuencialmente; el nombre que se da a cada una de ellas es MIpomoea_n, como se muestra en la Figura 29.

En el siguiente link se puede acceder a la descarga del código utilizado https://utneduec-my.sharepoint.com/:u:/g/personal/kfmoncayos_utn_edu_ec/EQL1iAfjbp5OlnSUDyBBI4IBx5NZBgGeNaCqP8ZLohH3NA?e=l0aDa3

Figura 28. Base de datos ordenadas

Fuente: Autoría propia

3.6.4.4. Etiquetado de imágenes

En esta etapa se etiqueta las imágenes con la ayuda del Software LabelImg, que es una herramienta gratuita de código abierto para etiquetar imágenes gráficamente. LabelImg está escrito en Python . Es una forma fácil y gratuita de etiquetar cientos de imágenes, además admite el etiquetado en formato de archivo de texto VOC XML o YOLO; en este caso se toma en cuenta el formato Yolo debido a que el sistema se lo realiza en Yolo.

Para realizar el proceso de etiquetamiento, la explicación se encuentra en el Anexo 7.

3.6.4.5. Entrenamiento mediante Redes Neuronales Convolucionales

Una vez adquiridas las imágenes en el cultivo de tomate riñón, y previamente etiquetadas con la herramienta LabelImg, se tiene archivos de cada imagen en una carpeta llamada labels. Los archivos que se producen son de tipo (.txt), los cuales contienen las coordenadas de la imagen en la que se encuentra la maleza Ipomoea; cabe recalcar que el formato que produce la herramienta de LabelImg, guarda los archivos en un formato admitido por Yolo.

Una vez que se tiene las imágenes respectivamente con el formato (.txt), se deben colocar en la carpeta data, esta carpeta se encuentra en la raíz de Darknet. Por otra parte, se debe crear dos carpetas, una que contiene el conjunto de imágenes a ser reconocidas y otra en la que contendrá los pesos y las ubicaciones de las imágenes para crear el dataset para el entrenamiento.

Finalizado este proceso; se descarga de la página oficial de Darknet Yolo Tiny V3 los pesos para el modelo de entrenamiento que se tiene.

Una vez se haya descargado el archivo, se da paso al entrenamiento en Colab. La Figura 30 muestra el proceso de entrenamiento del modelo.

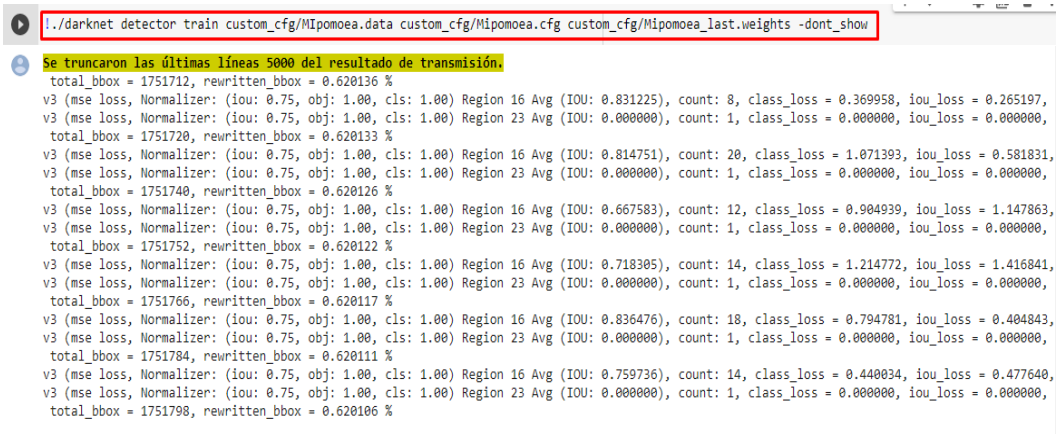

Figura 29. Proceso de entrenamiento del modelo

Fuente: Autoría propia

3.6.4.5.1. Parámetros de configuración en la red

Para el entrenamiento de la red, existen algunos parámetros que se deben configurar, a continuación, se detalla cada uno de ellos.

- Tamaño de la imagen: se refiere al tamaño de la imagen que procesa la red; el mismo que será fijo ya que encaja con el resto de la red y este es de 416 pixeles como se puede observar en la Figura 31. Todas las imágenes que pasen serán redimensionadas antes

de entrar a la red; ya que, si las imágenes son demasiado grandes, se necesita mayor procesamiento para el entrenamiento

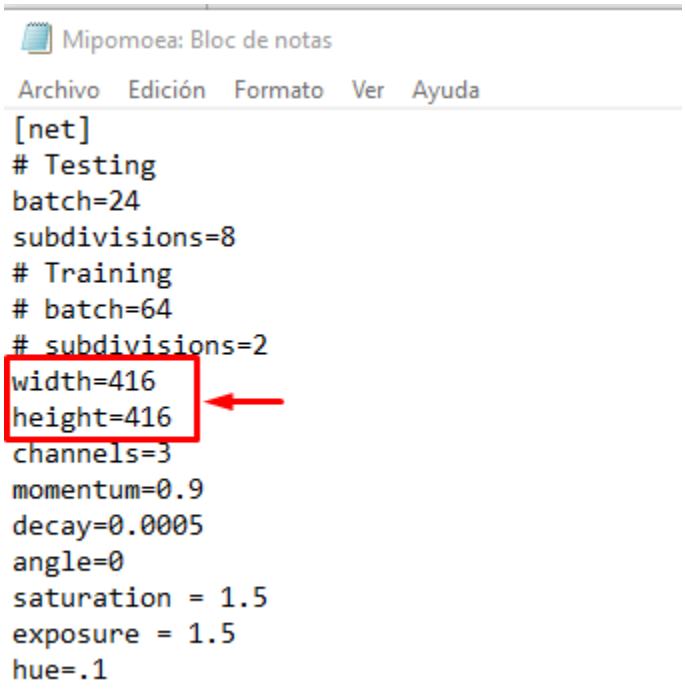

Figura 30. Configuración del tamaño de la imagen en la red

Fuente: Autoría propia

- Cantidad de cajas por imagen: Las cuales serán la cantidad de objetos máximos que queremos detectar, como se muestra en la Figura 32.

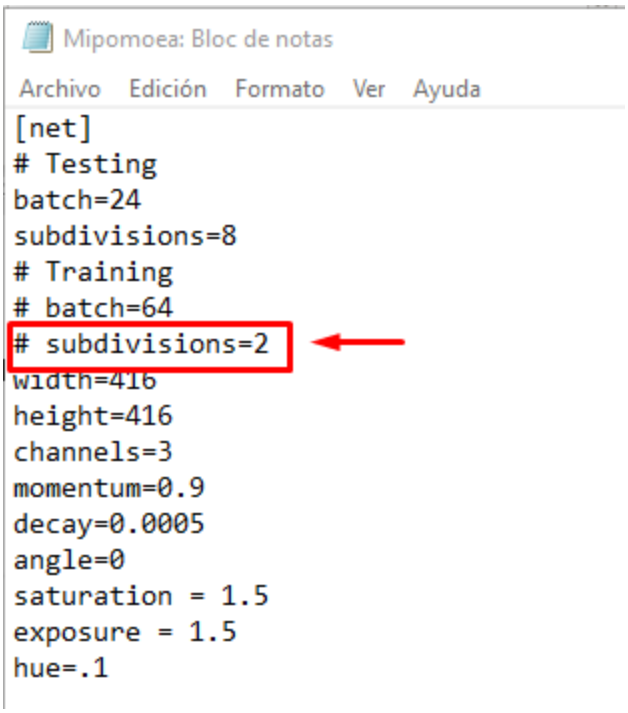

Figura 31. Configuración de la cantidad de cajas por imagen

Fuente: Autoría propia

- Etiquetas: serán los objetos que se desea detectar, en el sistema de reconocimiento realizado se detectará un tipo de maleza (MIpomoea); cómo se observa en la Figura 33.

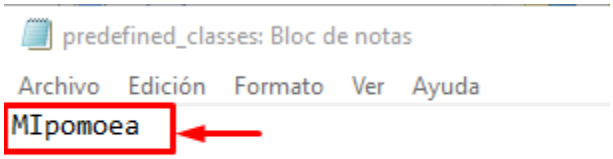

Figura 32. Configuración de las etiquetas

Fuente: Autoría propia

- *Epochs:* Es la cantidad de iteraciones sobre todo el dataset que realizará la red neuronal para su entrenamiento, en este caso se realizó 48000 *epochs*.

- *Train_times*: Este valor es la cantidad de veces de entrenar una misma imagen. Como se puede visualizar en la Figura 34. la información después de cada iteración, en este caso la 8122 donde ha analizado 194,928 imágenes:

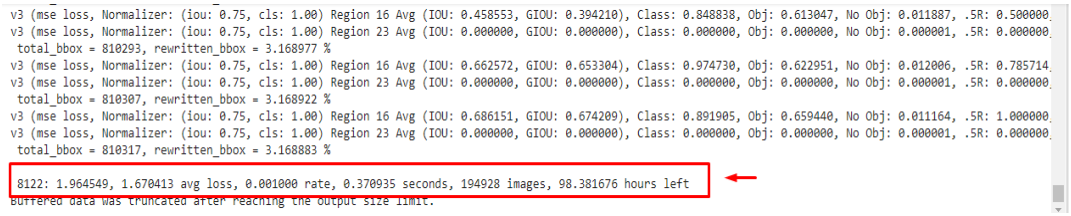

Figura 33. Configuración del train_times

Fuente: Autoría propia

- Saved_weights_name: Una vez que se haya entrenado la red, se debe guardar sus pesos en el mismo archivo y se lo usará para realizar las predicciones. El archivo weights se lo muestra en la Figura 35.

```
## Configuramos nuestra Red
modelConfiguration = "config_file/tomate.cfg"     #Archivo de arquitectura de la Red
modelWeights = "config_file/tomate.weights" # Pesos de la Red
net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights) # Configuracion de la red
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV) # configuramos opencv por detras
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)      # configuramos que trabaje con cpu
```

Figura 34. Archivo weights

Fuente: Autoría propia

3.6.4.5.2. Operaciones principales de las redes neuronales

Las operaciones principales de las redes neuronales se describen a continuación:

Se tiene una entrada, en este caso son las imágenes adquiridas en los cultivos de tomate riñón de invernadero, como se muestra en la Figura 36. un ejemplo de las fotos que se tomaron para crear el dataset a utilizar en el entrenamiento de la red.

Figura 35. Fotografías adquiridas para la construcción del dataset

Fuente: Autoría propia

Al iniciar, la red toma como entrada los pixeles de la imagen. En el caso de los datos utilizados para el presente proyecto, se tiene imágenes que oscilan entre los 960×720 pixeles de alto y ancho, esto equivale a utilizar 691200 neuronas. Pero esto sería en el caso de tener 1 sólo color es decir en escala de grises. En este caso, las imágenes son a color, se necesitaría 3 canales RGB (red, green, blue), quedando 960x720x3 = 2073600 neuronas. Estas neuronas constituyen nuestra capa de entrada.

El preprocesamiento en las redes neuronales convolucionales se lo realiza internamente, es decir, antes de alimentar la red se deben convertir los valores entre 0 y 1; por lo tanto, se divide todo entre 255, el valor de 255 es porque los colores de los pixeles tienen valores que van del 0 al 255.

Seguido de preprocesamiento, se realiza el procesado distintivo de las redes neuronales convolucionales, es decir, se realiza las convoluciones. Las cuales consisten en tomar grupos de pixeles cercanos de la imagen de entrada, y se va operando matemáticamente el producto escalar

con la matriz que se tiene(kernel). Con el tamaño del kernel se logra visualizar todas las neuronas de entrada de izquierda-derecha, de arriba-abajo; con esto se genera una nueva matriz de salida, la cual será la nueva capa de neuronas ocultas. Como se muestra en la Figura 37 la imagen de entrada y en la Figura 38. el tamaño de kernel de dicha imagen.

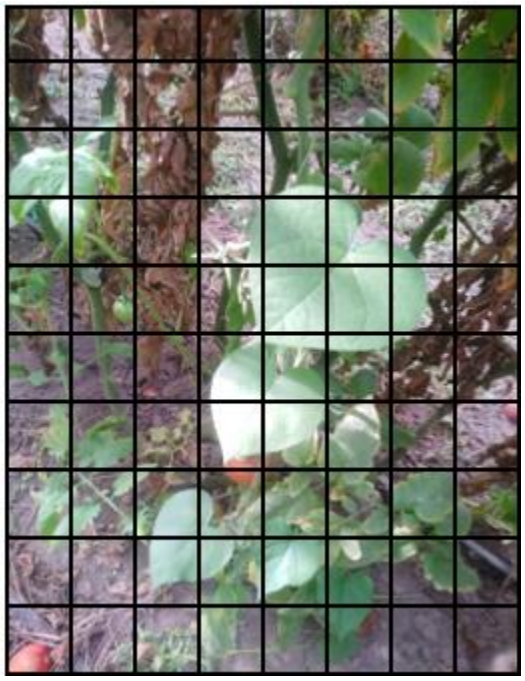

Figura 36. Imagen de entrada

Fuente: Autoría propia

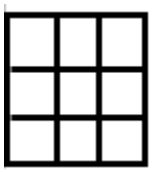

Figura 37. Kernel

Fuente: Autoría propia

El kernel toma valores aleatorios, para hacer que siga la distribución normal.

Seguidamente se tiene el filtro (conjunto de kernels); en este caso no se aplicará un sólo kernel, sino que se tendrá muchos kernel, a los cuales se los llama filtros. En nuestro caso se tiene

256 filtros como se puede observar en la Figura 39.; con lo cual se tendrá 256 matrices de salida, lo que se conoce como *feature mapping*.

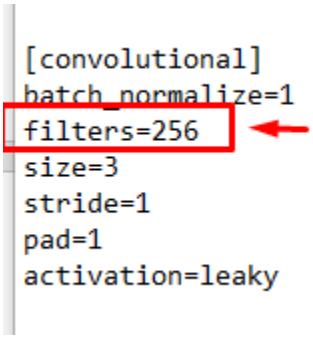

Figura 38. Filtros aplicados en la primera capa oculta de neuronas

Fuente: Autoría propia

Cada una de 256x256x1, dando un total de 65536 neuronas para nuestra primera capa oculta de neuronas. En la figura 40 se ve un ejemplo del kernel que realiza el producto matricial con la imagen de entrada y se desplaza1 pixel de izquierda a derecha y de arriba-abajo; con esto va generando una nueva matriz que compone el mapa de features.

Figura 39. Kernel que realiza el producto matricial con la imagen de entrada

Fuente: Autoría propia

Conforme se desplaza el kernel, se obtiene una nueva imagen filtrada por el kernel. Como se observa en la primera convolución es como que se tuviera 256 imágenes filtradas nuevas; las cuales están dibujando ciertas características de la imagen original, lo que ayudará más adelante a poder distinguir un objeto de otro, por ejemplo, la maleza Ipomoea. En esta red neuronal convolucional se utilizó la función de activación que es la *ReLu* (*Rectifier Linear Unit*).

Siguiendo con las operaciones que se realiza en las redes neuronales convolucionales se tiene el muestreo o *subsampling*[8]*;* en este paso, antes de hacer una nueva convolución, se toma una muestra de las neuronas más representativas. El muestreo se lo realiza para reducir el tamaño de la próxima capa de neuronas, de esta manera se conserva las características más importantes que detectó cada filtro. Existen diversos tipos de muestreo; de los cuales se utilizó el *max-pooling*.

Al utilizar la técnica de max-pooling con un tamaño de 2×2, como se muestra en la Figura 41(tamaño que se asigna en las configuraciones de la red). Lo que quiere decir que sólo se tomará el tamaño asignado y se irá preservando el valor más alto de los 4 pixeles; es decir la imagen es reducida a la mitad y quedará 128 x 128 pixeles. Con esta técnica se almacena la información más importante para detectar características deseadas.

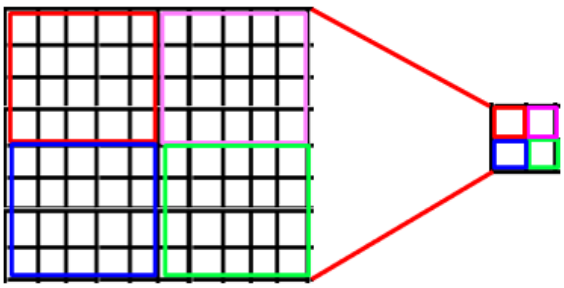

Figura 40. Aplicando técnica de *max-pooling* 2x2

Fuente: Autoría propia

[8] Es una técnica con la cual se comprime la información de color que contiene una señal para favorecer a la información contenida en la luminancia.

Una vez aplicada la técnica del *max-pooling*, se realiza la primera convolución; en la Figura 42 se observa la representación de la primera convolución, que consiste en una entrada y un conjunto de filtros, con ello se genera un mapa de características y se hace un submuestreo. Tomar en cuenta que el resultado es para tres capas, debido a que las imágenes son a color.

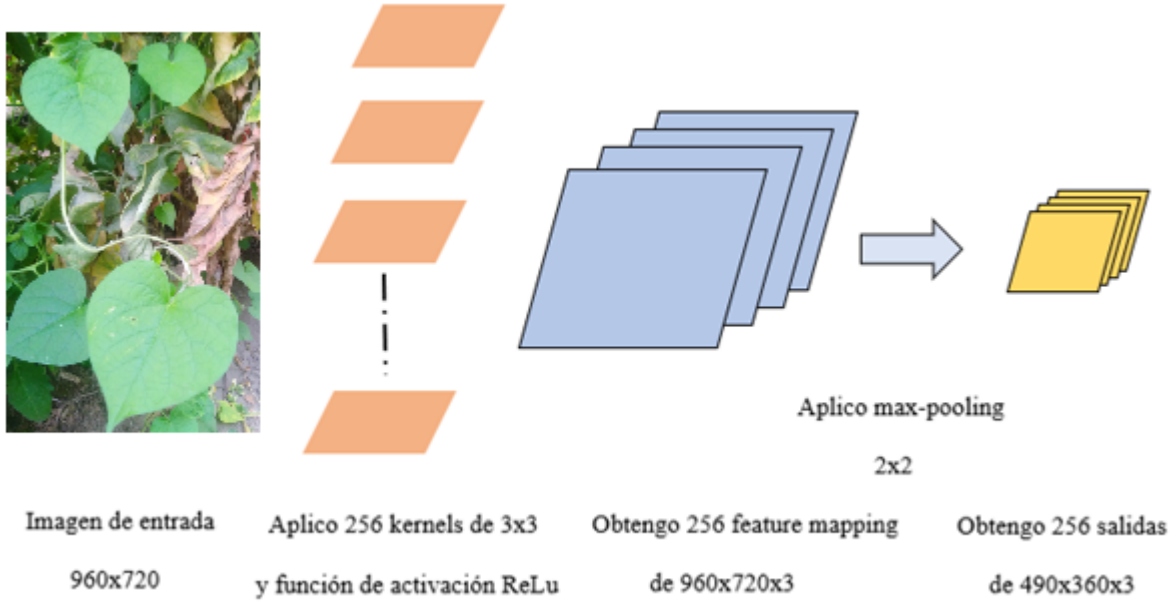

Figura 41. Representación de la primera convolución

Fuente: Autoría propia

En la primera convolución se detecta las características primitivas, es decir: las líneas o curvas. Conforme se vaya realizando más capas con las convoluciones, los mapas de las características serán capaces de reconocer formas más complejas, dando como resultado reconocimiento deseado. En la Tabla 11. Se muestra los datos obtenidos con la primera convolución.

Tabla 11. Tabla de datos obtenidos con la primera convolución

Primera Convolución	Se aplica Kernel	Se obtiene Feature Mapping	Se aplica Max-pooling	Se obtiene la salida de la primera Convolución
960x720x3 = 2073600 neuronas	256 filtros de 3×3	960x720 x 256 kernel = 176947200 neuronas	de 2×2	490x360x3 = 529200 neuronas

Fuente: Autoría propia

Si continuamos con las convoluciones subsecuentes, en esta etapa se realizará una segunda convolución como se indica en la Figura 43< mientras que en la Tabla 12 se muestra los datos obtenidos con la segunda convolución.

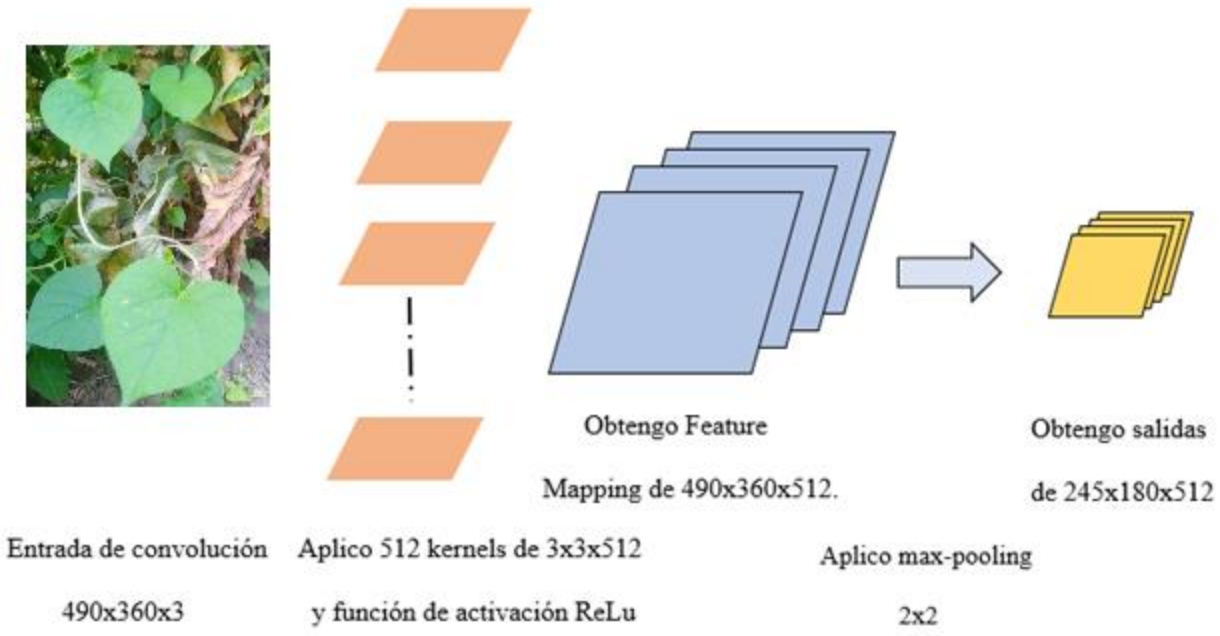

Figura 42. Representación de la segunda convolución

Fuente: Autoría propia

Tabla 12. Tabla de datos obtenidos con la segunda convolución

Segunda Convolución	Se aplica Kernel	Se obtiene Feature Mapping	Se aplica Max-pooling	Se obtiene la salida de la primera Convolución
490x360x3 = 529200 neuronas	512 filtros de 3×3	490x360x 512 kernel = 90316800 neuronas	de 2×2	245x180x512 = 22579200 neuronas

Fuente: Autoría propia

Como se puede observar, se realizan algunas convoluciones, en donde se siguen los pasos anteriormente realizados, hasta llegar a la última convolución.

3.6.4.6. Reconocimiento de la maleza Ipomoea

Una vez que el modelo creado en Yolo Tiny V3 ha sido entrenado mediante Darknet, ya se puede realizar las pruebas para realizar el reconocimiento de la maleza Ipomoea en cultivos de tomate riñón, como se puede observar en la Figura 44.

Figura 43. Maleza Ipomoea reconocida utilizando el modelo entrenado

Fuente: Autoría propia

En la Figura 44 se observa la manera en que, el modelo entrenado puede reconocer la maleza en tiempo real. La maleza Ipomoea o campanilla reconocida es etiquetada con un cuadro delimitador y el nombre de la categoría a la cual pertenece. Algo muy importante, es que, si existen más malezas de la misma categoría será reconocida y etiquetada. Cabe recalcar que mediante *Darknet,* permite realizar el reconocimiento de la maleza Ipomoea en tiempo real, mediante una imagen o a su vez mediante un video.

Para el entrenamiento de la red, se tiene Yolo Tiny V3 de la interfaz darknet; en el archivo de *custom.cfg*, que es el archivo de configuración en el cual se establecen los parámetros como las capas de la red neuronal, clases, filtros, entre otras, de acuerdo con las necesidades del usuario. Después de completar el entrenamiento con el archivo de pesos generados, se procede a realizar las respectivas pruebas de detección de objetos.

CAPÍTULO IV. PRUEBAS DE FUNCIONAMIENTO.

En este capítulo se procede a realizar las pruebas respectivas del sistema de reconocimiento de la maleza Ipomoea, desarrollado en el capítulo III. Para lo cual se utilizó en la adquisición de datos la cámara de un teléfono celular y como se había mencionado las fotografías adquiridas no deben ser de alta resolución. Para las pruebas de funcionamiento se dispone de una cámara Raspberry rev 1.3 de 5 megapíxeles. De igual manera se toma en consideración al objeto a reconocer (maleza Ipomoea) con dicho objeto se procede a realizar las pruebas de funcionamiento a escala de laboratorio, mediante la adquisición de la maleza Ipomoea para su respectivo reconocimiento y posterior a eso se presenta los resultados y conclusiones.

4.1. Adquisición de datos

Al trabajar con Redes Neuronales Convolucionales se observa que el número de datos es importante, ya que, mientras más datos se tenga, el sistema es más eficiente; pero de igual manera se necesita mayor procesamiento para su entrenamiento. Al trabajar en el entorno de colab 9se dispone de 12 horas de corrido para entrenar en la nube, el código en donde se realizó el entrenamiento del sistema se encuentra en el siguiente link: https://colab.research.google.com/drive/1KCA4R3SPR49AY51SJuxvRiZvQCOG8O0C?usp=sharing.

En la toma de fotografías se realiza varias pruebas, en las que se puede observar que, la dimensión óptima para el entrenamiento es de 960*720 pixeles como se muestra en la Figura 45. En un inicio se tomaron datos de alta calidad y como resultado se obtuvo imágenes muy grandes y el sistema se demoró más tiempo en su entrenamiento.

[9] Entorno gratuito de Jupyter Notebook, no requiere configuración y se ejecuta en la nube. Colab, dispone de acceso gratuito a la informática de GPU y TPU; el cual permite trabajar y desarrollar algoritmos.

El número de datos que se consideró es de 925 imágenes; las cuales se las enumeró con un nombre específico y ordenado para su etiquetamiento.

Figura 44. Adquisición de imágenes

Fuente: Autoría propia

4.2. Pruebas del sistema

En las pruebas a realizarse, previamente se considera si el sistema va a ser implementado en algún sitio fijo o si se va a realizar pruebas a escala de laboratorio. En este caso las pruebas se las realiza a escala de laboratorio; por lo tanto, se tiene una estación de trabajo como se muestra en la Figura 46.

Figura 45. Estación de trabajo

Fuente: Autoría propia

Para lo cual, como se había mencionado anteriormente, se adquiere tanto imágenes, como videos de la maleza Ipomoea; las cuales facilitarán las pruebas del sistema de reconocimiento si se lo realiza mediante imágenes o videos guardados. A continuación, se detalla el diagrama de conexión del sistema que se utilizó.

4.2.1. Diagrama de conexión basado en la arquitectura del sistema.

En este apartado se unifica todos los elementos del sistema. Es por ello, que se considera el diagrama del sistema que se muestra en la Figura 47., donde se tiene que: mediante el raspberry pi 4 tiene dos conexiones, una directamente conectado a la cámara raspberry pi y otra conectada directamente al monitor en donde se visualiza el reconocimiento de la maleza Ipomoea.

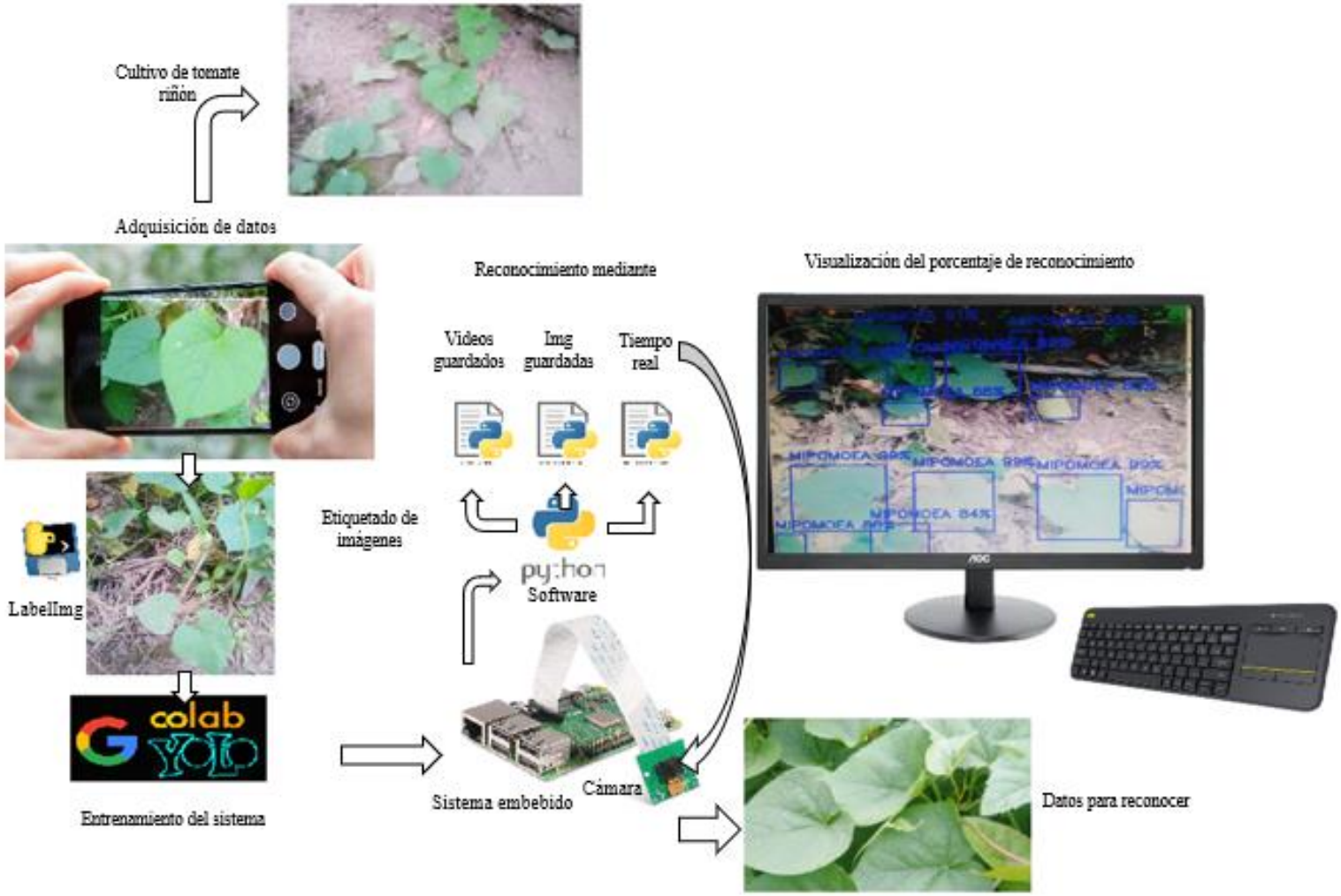

Figura 46. Diagrama de bloques basado en la arquitectura del sistema

Fuente: Autoría propia

Al ingresar y poner en marcha nuestro sistema de reconocimiento, se tiene ubicada la carpeta del sistema en la carpeta de Documentos; donde se encuentra la subcarpeta llamada Detección_MIpomoea; dentro de la misma se tiene las carpetas que contienen los archivos completos del sistema, como se puede visualizar en la Figura 48.

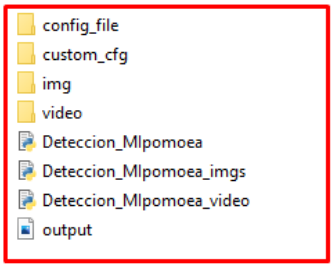

Figura 47. Sistema de Detección de tomates

Fuente: Autoría propia

En el Anexo 8, se detalla cada uno de los elementos que se tiene dentro de las configuraciones del sistema.

Al considerar que la pruebas que se realizan, son a escala de laboratorio, se dispone del sistema de reconocimiento, que se lo puede comprobar de tres formas:

- Mediante imágenes guardadas
- Mediante un video tomado
- En tiempo real

A continuación, se detalla las pruebas que se realiza con cada una de ellas.

4.2.1. Pruebas del sistema en tiempo real

Para realizar las pruebas del sistema en tiempo real, se tiene el código mencionado en el Anexo 8, con el cual se abre el código, lo ejecutamos y se procede a visualizar y detectar la maleza

Ipomoea. En la Figura 49. Se muestra el reconocimiento de la maleza Ipomoea, junto al porcentaje de precisión y etiqueta dada.

Figura 48. Reconocimiento de la maleza Ipomoea en tiempo real

Fuente: Autoría propia

En la Figura 50, se observa que, se ha tomado de ejemplo, algunas malezas de tipo Ipomoea; con lo cual se verifica el nivel de precisión que se tiene; como resultado se obtiene un reconocimiento óptimo en los cuales se indica con cuadros delimitadores la maleza reconocida por el sistema.

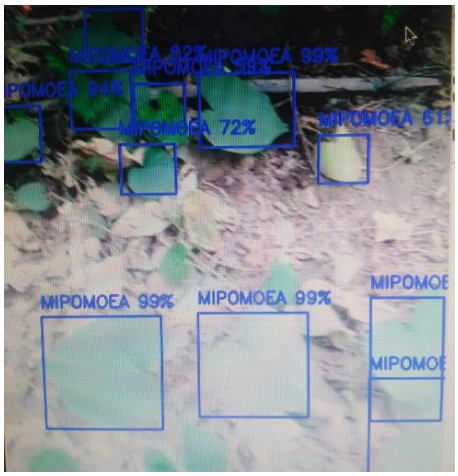

Figura 49. Reconocimiento de la maleza Ipomoea en tiempo real

Fuente: Autoría propia

Se toma, otras muestras para verificar la precisión del sistema en tiempo real, en la Figura 51., se verifica el porcentaje de precisión en cuanto a su forma, de igual manera se observa que, si se tiene malezas juntas, de igual manera las detecta.

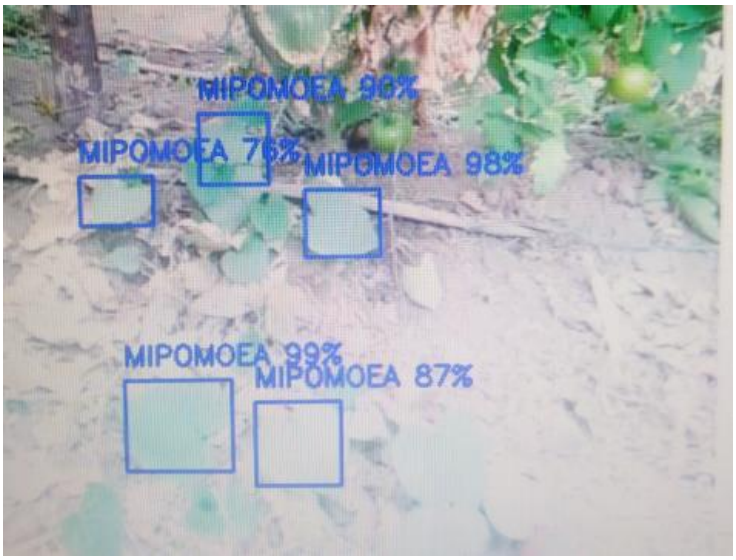

Figura 50. Reconocimiento de la maleza Ipomoea en tiempo real

Fuente: Autoría propia

4.2.2. Pruebas del sistema mediante imágenes guardadas

Para realizar las pruebas del sistema mediante imágenes guardadas, se tiene el código mencionado en el Anexo 8, con el cual se abre el código, y, para que la imagen guardada sea reconocida, se debe ingresar el nombre de la imagen y su extensión en el bloque de configuración de la red, como se muestra en la Figura 52.

```
## Configuramos nuestra Red
modelConfiguration = "config_file/MIpomoea.cfg"     #Archivo de arquitectura de la Red
modelWeights = "config_file/Mipomoea_30000.weights" # Pesos de la Red
net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights) # Configuracion de la
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV) # configuramos opencv por detras
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)      # configuramos que trabaje con cpu

############################################
cv.namedWindow("Input", cv.WINDOW_AUTOSIZE)
cv.namedWindow("Output", cv.WINDOW_AUTOSIZE)

img = cv.imread("img/MIpomoea_13.png")

cv.imshow('Input', img)
#cv.waitKey(0)

blob = cv.dnn.blobFromImage(img, 1 / 255, (whT, whT), [0, 0, 0], 1, crop=False)
net.setInput(blob)
layersNames = net.getLayerNames()
outputNames = [(layersNames[i[0] - 1]) for i in net.getUnconnectedOutLayers()]
outputs = net.forward(outputNames)
findObjects(outputs,img)

cv.imwrite("output.png",img)

cv.imshow('Output', img)
cv.waitKey(0)
cv.destroyAllWindows()
```

Figura 51. Ingreso del nombre de la imagen a ser reconocida y su extensión

Fuente: Autoría propia

Una vez ingresados los datos mencionados, se ejecuta el código que se indica en el Anexo 8., con lo cual se puede observar el resultado en la Figura 53.

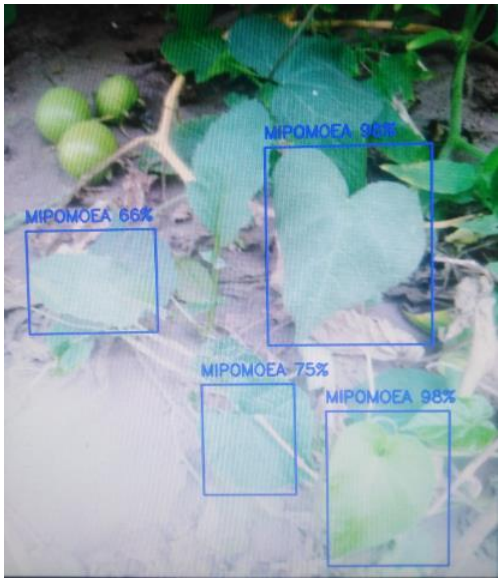

Figura 52. Reconocimiento de la maleza Ipomoea mediante imágenes guardadas

Fuente: Autoría propia

Para realizar estas pruebas se ha guardado imágenes tomadas en el cultivo de tomate riñón, las cuales contienen maleza Ipomoea, como se muestra a continuación:

El resultado del reconocimiento de maleza Ipomoea se indica en la Figura 54.

Figura 53. Imágenes guardadas de la maleza Ipomoea

Fuente: Autoría propia

El resultado del reconocimiento de la maleza Ipomoea se indica en la Figura 55.

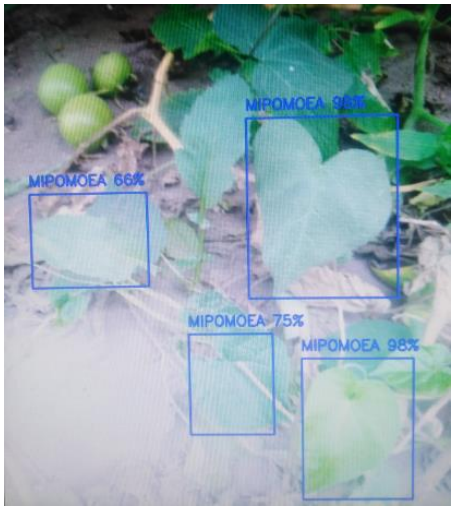

Figura 54. Imágenes guardadas de la maleza Ipomoea

Fuente: Autoría propia

4.2.3. Prueba del sistema mediante un video guardado

Para realizar las pruebas del sistema mediante un video guardado, se tiene el código en el Anexo 8, el mismo que, se abre y se ingresa el nombre del video y su extensión en el bloque de lectura del video

. Una vez que se haya modificado el nombre y la extensión, se ejecuta el código y se verifica que se realiza el reconocimiento, como se muestra en la Figura 56.

Figura 55. Reconocimiento de la maleza Ipomoea en un video guardado

Fuente: Autoría propia

En la Figura 56, se ha tomado como ejemplo un video grabado de la maleza Ipomoea dentro del cultivo de tomate riñón; con el que se verifica la precisión del sistema, al reconocer cada una de las malezas correctamente y con sus etiquetas correspondientes.

Mientras el video está siendo reproducido, se toma una captura para verificar que, se realiza correctamente el reconocimiento de la maleza, la cual se observa en la Figura 57.

Figura 56. Reconocimiento de la maleza Ipomoea mediante un video guardado

Fuente: Autoría propia

CONCLUSIONES Y RECOMENDACIONES.

5.1. Conclusiones

Con la fundamentación teórica acerca del reconocimiento de malezas utilizando visión artificial mediante redes neuronales, se concluyó que existen algunos tipos de redes neuronales con las que se puede trabajar, de las cuales se utilizó la más apropiada. En la realización del sistema actual se utilizó las redes neuronales convolucionales.

Conforme se observó en el apartado del marco teórico, se tiene diferentes tipos de malezas; por lo que para su reconocimiento se tomó una especie en sí (maleza Ipomoea o campanilla). Con dicha información acerca de las malezas se pudo concluir que, en el caso de este sistema, se escogió la maleza mencionada, por la forma que tiene y todas las características que posee, ya que de esta manera en su entrenamiento se diferenciaría de la planta y se obtendría un reconocimiento óptimo.

Con el entrenamiento mediante redes neuronales convolucionales se pudo verificar que las mismas nos ayuda a extraer características profundas de una imagen, en este caso se extrajo las características de la maleza Ipomoea; estas características disponen de numerosas capas ocultas en las que se descubre y reconoce diferentes patrones acordes de la maleza mencionada.

Se pudo concluir que, en las CNN un factor importante para que el modelo tenga precisión, es la adquisición de datos en grandes volúmenes, ya que las redes neuronales convolucionales tienen mayor éxito cuando disponen de grandes cantidades de datos.

Con la utilización de un sistema embebido para la ejecución del sistema, se llegó a la conclusión de que, es muy importante acorde a la red que se utilice y las características que posea el mismo. En este caso se utilizó un Raspberry Pi 4, debido a las características que se presentaron en la tabla comparativa realizada con un Raspberry pi Zero, el cual dispone de muy bajo procesamiento y no se lo podría utilizar en esta clase de proyectos.

Debido a que, en una red neuronal se tiene dos fases: una fase es para su entrenamiento y la segunda fase es para su validación; se concluye que, en las dos fases se utiliza datos distintos por lo que se debe realizar la toma de datos para el entrenamiento en el que se agrupa imágenes para la extracción de características y para su validación que es en donde se necesita imágenes para comprobar la red entrenada.

Por otro, el uso de la herramienta LabelImg es de gran importancia, en la etapa de etiquetamiento; ya que esta posee el formato YOLO que es compatible con la versión de Yolo Tiny V3, teniendo como resultado archivos (.txt), los cuales contienen información de cada una de las imágenes con las que se va a entrenar el sistema de reconocimiento.

Con la utilización de colab, se pudo concluir que, el mismo permitió ejecutar y programar en Python un navegador propio directo desde nuestro pc, ya que dispone de algunas ventajas como: no requiere de ninguna configuración, brinda acceso gratuito a GPUs y permite compartir contenido fácilmente; lo que brindó mayor facilidad para realizar el entrenamiento sin necesidad de contar con un dispositivo de alto procesamiento.

Se pudo comprobar que el Yolo Tiny V3 puede ser entrenado y dar resultados óptimos y precios en cuanto al reconocimiento; por otro lado, las pruebas realizadas se las puede hacer mediante imágenes, video o a su vez en tiempo real, lo cual no baja el rendimiento del sistema.

Al realizar una tabla comparativa entre Yolo V3 y Yolo Tiny V3 para la utilización en el sistema, se llegó a la conclusión de que, se tiene que en ambas redes ofrecen buena precisión; pero en el caso de Yolo Tiny V3 es hasta cinco veces más rápida que Yolo V3.

Mediante las pruebas de funcionamiento, se pudo verificar que, el sistema con la utilización de redes neuronales convolucionales ha sido exitoso debido a la utilización de Yolo Tiny V3, la

cual ha mostrado tener una buena precisión; cabe recalcar que mientras más datos se tenga y mayor epochs tenga en su entrenamiento, el sistema va a tener más precisión

Al realizar las pruebas y comprobar los resultados deseados en el reconocimiento de malezas, se puede afirmar que se ha logrado desarrollar un sistema capaz de reconocer los objetos esperados.

5.2. Recomendaciones

Es recomendable que realice la investigación teórica necesaria para saber que herramientas de hardware y software se necesita para la creación del sistema deseado, así como también con este estudio previo se tendrá un conocimiento general de la teoría, de esta manera poder lograr los resultados deseados.

Es importante conocer y analizar la arquitectura y el funcionamiento que tiene una red neuronal; con el fin de comprender su funcionamiento y saber los tipos de redes neuronales existentes, para posteriormente escoger la red con la que se va a realizar el proyecto.

Al realizar la investigación necesaria con las arquitecturas de las redes neuronales convolucionales, es recomendable utilizar una arquitectura que no disponga de muchos recursos computacionales para que el rendimiento del sistema sea el adecuado.

Es recomendable que si no se dispone de un equipo que posea altos recursos de cómputo que tenga una CPU de última generación o GPU que tenga un procesamiento dedicado; tenga la opción de poder utilizar desde su computador normal en un navegador la herramienta de Google Colaboratory; el cual brinda acceso gratuito a GPUs y permite compartir contenido fácilmente.

Es recomendable adquirir gran cantidad de datos para el entrenamiento de la red, ya que mientras más datos ingresen a la red, el sistema aprende más características del objeto seleccionado en la imagen, aumentando la precisión de la detección.

Es importante tomar en cuenta que las imágenes que se adquiera no deben estar tomadas en la más alta calidad, ya que se vuelve muy pesado el entrenamiento; es recomendable utilizar dimensiones estándares.

Una recomendación para la captura de imágenes: si se va a utilizar la cámara de un móvil, conviene hacer fotos con pocos megapíxeles, pues si hace una imagen de 4K de 5 Megas, luego la

red neuronal la reducirá a 416 pixeles de ancho, por lo que se tendrá un coste adicional de ese preprocesamiento en tiempo, memoria y CPU.

Dentro de lo que es redes neuronales se mencionan capas densas o totalmente conectadas, lo cual es un funcionamiento muy interesante, aunque estás no son ideales para el procesamiento trabajar con imágenes. Para trabajar con imágenes es recomendable y es más eficiente, trabajar con redes neuronales convolucionales.

En el etiquetado de las imágenes, si se va a trabajar con Yolo en cualquiera de sus versiones, es recomendable utilizar la herramienta LabelImg, puesto que la misma dispone del formato YOLO que es compatible con nuestra red; guardando archivos (.txt). Al ser compatible en los formatos, nos ayuda de una manera más eficiente en el entrenamiento.

Al elegir la versión de Yolo, tomar muy en cuenta con el dispositivo embebido a utilizarse, ya que depende las características que tenga el mismo puede ser más eficiente una u otra versión de Yolo.

Es recomendable que, al momento de realizar las pruebas de funcionamiento, se las realice mediante imágenes guardadas, mediante un video y en tiempo real; de esta manera se podrá comprobar el nivel de precisión que tiene el sistema y se podrá observar que no se ralentiza el sistema.

Anexo 1: Entrevista

Sr. Wilson Suárez

Agricultor y encargado, cultivos de tomate riñón “TOLAS SOCAPAMBA”

“Los cultivos de tomate riñón al momento de salir de los invernaderos las plantas pequeñas y se las planta directo en el invernadero, el momento del riego salen las malezas que existen. Las malezas nacen todos los días, debido a la humedad, es por ello que se extraen con pala cada 15 días, ya que si no se las saca se pierde la producción debido a las enfermedades que pueden nacer en dicho cultivo. Para nosotros los agricultores si es factible el disponer de un sistema actual para el reconocimiento de malezas, ya que ahorraría tiempo y dinero”

Anexo 2: Encuesta

Preguntas de la Encuesta

Pregunta 1

¿Cada que tiempo crecen las malezas?

1. Siempre
2. Cada semana
3. Nunca

Pregunta 2

Acorde al crecimiento de malezas dentro de los cultivos de tomate riñón, ¿Cada que tiempo se saca la maleza?

1. Todos los días
2. Cada 15 días
3. Nunca

Pregunta 3

Al detectar la maleza dentro del cultivo de tomate riñón, la extracción de la misma ¿De qué manera se la realiza?

1. Tradicional
2. Otro (indique cual)

Pregunta 4

Si no se extrae a tiempo la maleza dentro del cultivo de tomate riñón de invernadero ¿Qué sucede?

1. Nada

2. Enferma y pierde la producción

Pregunta 5

¿El implementar un sistema de reconocimiento de malezas de tomate riñón ayuda en cuanto a tiempo y dinero a los agricultores?

1. Si
2. No

Anexo 3: Respuestas de la Encuesta

Preguntas de la Encuesta

Tabla 13. Tabla de Respuestas de la Encuesta Pregunta 1.

Número	Siempre	Cada semana	Nunca
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
Total	100%	0%	0%

Fuente: Autoría propia

Mediante esta pregunta se pudo determinar dentro de un cultivo de tomate riñón las malezas siempre van a existir por lo que es necesario dichas malezas por lo tanto puede ser factible realizar un sistema que solvente este problema.

Tabla 14. Tabla de Respuestas de la Encuesta Pregunta 2.

Número	Todos los días	Cada 15 días	Nunca
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0
Total	0%	100%	0%

Fuente: Autoría propia

Mediante esta pregunta se pudo determinar que se saca las malezas cada 15 días por lo que sería factible implementar un sistema de detección de malezas, el cual ayudaría a detectarlas inclusive en menos tiempo.

Tabla 15. Tabla de Respuestas de la Encuesta Pregunta 3.

Número	Tradicional	Otros
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
Total	100%	0%

Fuente: Autoría propia

Mediante esta pregunta se pudo determinar que la extracción de malezas, se la realiza de forma tradicional, por lo que un sistema de detección ayudará a estar más pendiente de la existencia de las misma, optimizando el tiempo de los agricultores al momento de la visita para extraer malezas cada cierto.

Tabla 16. Tabla de Respuestas de la Encuesta Pregunta 4.

Número	**Nada**	**Enferma y pierde la producción**
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
Total	0%	100%

Fuente: Autoría propia

Mediante esta pregunta se puedo determinar que, si no se extrae las malezas del cultivo de tomate riñón a tiempo, estas se enferman y se pierde la producción; por lo que se comprueba que, si es factible implementar un sistema de detección de malezas, de esta manera se logra no perder la producción por enfermedades debido al crecimiento de malezas.

Tabla 17. Tabla de Respuestas de la Encuesta Pregunta 5.

Número	**Si**	**No**
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
Total	100%	0%

Fuente: Autoría propia

Mediante esta pregunta se puedo determinar que al implementar un sistema de reconocimiento de malezas de tomate riñón ayuda en cuanto a tiempo y dinero a los agricultores, lo que brindaría ventajas a los mismos.

Anexo 4: Lenguajes de programación

1. Lenguaje de Programación Python

El lenguaje de programación Python es un lenguaje de alto nivel, el cual permite procesar todo tipo de estructuras de datos.

Características:

- Una de las características principales de Python es que es un lenguaje interpretado, ya que su interprete traduce instrucción por instrucción al código escrito.
- Python es multiplataforma, que quiere decir que su código fuente se vuelve ejecutable para varios sistemas operativos.
- El lenguaje de programación Python también tiene como característica de que es multiparadigma, que quiere decir que soporta varios paradigmas de programación.
- Python es un software libre y es gratuita su descarga
- Dispone de extensas y potentes librerías
- Python es un lenguaje de programación de propósito general.

En esta lección se busca introducir al lenguaje Python, sus características, modos de instalación, soporte comunitario, y los recursos más destacados disponibles en la Web para tomar en cuenta. A continuación, el temario de esta lección:

2. Matlab

MATLAB es un entorno es un entorno completo en donde los problemas y sus soluciones se expresan de una manera sencilla, sin tener la necesidad de hacer uso de la programación tradicional. De igual manera es un entorno de computación y desarrollo de aplicaciones, las cuales

se encuentran integradas para realizar proyectos que tengan que ver con cálculos matemáticos y la visualización gráfica de los mismos.

Características:

- Matlab es un lenguaje de alto nivel, el cual es utilizado para cálculos científicos y de ingeniería
- Dispone de un entorno de escritorio optimizado para la exploración, el diseño y la solución de problemas
- Matlab dispone de gráficas para visualizar todos los datos y las herramientas para crear diagramas personalizados.
- Tiene aplicaciones para ajustar curvas, analizar señales, ajustar sistemas de control, clasificar datos, ente otras tareas
- Matlab tiene herramientas para crear aplicaciones con interfaces personalizadas de usuario
- Tiene Interfaces para Java, C/C++, .NET, SQL, Python, y Microsoft® Excel®
- Matlab tiene implementación libre de derechos para compartir programas de MATLAB con los usuarios finales

Tabla 18. Tabla de Comparación entre Python y Matlab

Base de comparación entre Python y Matlab	Python	Matlab
Definición	Matrices numéricas y tipo de datos Un lenguaje de programación de propósito general de alto nivel	Lenguajes orientados a matemáticas y matrices MATLAB es el lenguaje de alto rendimiento para informática técnica
Uso	Python se puede utilizar para programación web (Zope, Google App Engine y mucho más)	MATLAB permite manipulaciones matriciales, trazado de funciones y datos, creación de interfaces de usuario
Beneficios	Amplias bibliotecas de soporte. Desarrollo comunitario y de código abierto.	Matlab le permite probar algoritmos inmediatamente sin el acto de compilar,
Actuación	Álgebra lineal, gráficos y estadísticas de alto rendimiento. Llamadas de biblioteca optimizadas	El rendimiento mejorado requiere instalar, compilar, validar y adoptar complementos orientados al desarrollador
Académica	Fue desarrollado por la fundación de software Python en el año 1991.	La versión básica de Matlab está en el mercado desde la década de 1970.
Biblioteca	Consiste en una extensa biblioteca estándar	La biblioteca estándar no contiene funciones de programación genéricas.
Tiempo real Apoyo	Soporte personalizado por correo electrónico y teléfono	Sin soporte personalizado en tiempo real
Generación de código	Sin código completo y automático Generación para sistemas embebidos.	El código MATLAB genera código c y c ++ legible y portátil.

Fuente: https://www.educba.com/python-vs-matlab/

Anexo 5: Algoritmo

- **YoloV3**

YOLOv3 predice cajas en 3 escalas diferentes. Nuestro sistema extrae características de esas escalas utilizando un concepto similar para presentar redes piramidales [8]. Desde nuestro extractor de características base agregamos varias capas convolucionales. El último de estos predice un cuadro delimitador de codificación de tensor 3-d, objetividad y predicciones de clase.

Características:

- YOLO v3 usa una modificación de Darknet, la cual inicialmente posee una red de 53 capas entrenadas en Imagenet.
- Para la detección, 53 capas adicionales se apilan en ella, lo que resulta en una arquitectura subyacente totalmente convolucional de 106 capas para YOLO v3. Esta es la causa principal que ralentiza a YOLO v3 (30 FPS).
- Requiere más iteraciones para obtener una alta precisión (mAP),
- **YoloV3 Tiny**

YoloV3 Tiny, es un algoritmo de detección de objetivos liviano aplicado a plataformas integradas basadas en YOLOv3. YoloV3 Tiny redujo la red de detección de características YOLOv3 darknet-53 a una convolución tradicional de 7 capas y una capa de agrupación máxima de 6 capas, utilizando una red de predicción de dos escalas 13 * 13, 26 * 26 para predecir el objetivo.

Características:

- YOLOv3 Tiny es hasta cinco veces más rápida que YOLOv3.Esta es la causa principal que ralentiza a YOLO v3 (30 FPS).

- YoloV3 Tiny, tiene una velocidad de inferencia muy notable y superando en precisión al resto
- Requiere menos iteraciones para obtener una alta precisión (mAP), a comparación de YoloV3

Tabla 19. Tabla de Comparación entre Python y Matlab

Base de comparación entre YoloV3 y YoloV3 Tiny	YoloV3	YoloV3 Tiny
FPS	4.4 FPS	32 FPS
mAP	Más iteraciones para tener una alta precisión	Requiere menos iteraciones para tener una alta precisión
Capas de convolución	106	26

Fuente: Autoría propia

Anexo 6: Sistema Embebido

Placas eléctricas

- Raspberry Pi Zero

La Raspberry Pi Zero W tiene un diseño adecuado para el desarrollo de herramientas educativas y proyectos en los que se requieren un ordenador, esta placa dispone de tecnologías inalámbricas tales como: Wi-Fi 802.11b/g/n y Bluetooth BLE. La Raspberry Pi Zero es pequeña, a pesar de su tamaño tiene la misma capacidad que un Raspberry Pi 1 B, con la diferencia que al ser muy pequeña no incluye interfaces de Ethernet, USB, pero si dispone de una interfaz mini HDMI para video de alta resolución.

Se puede apreciar en la Figura 58 todos los componentes que dispone la placa.

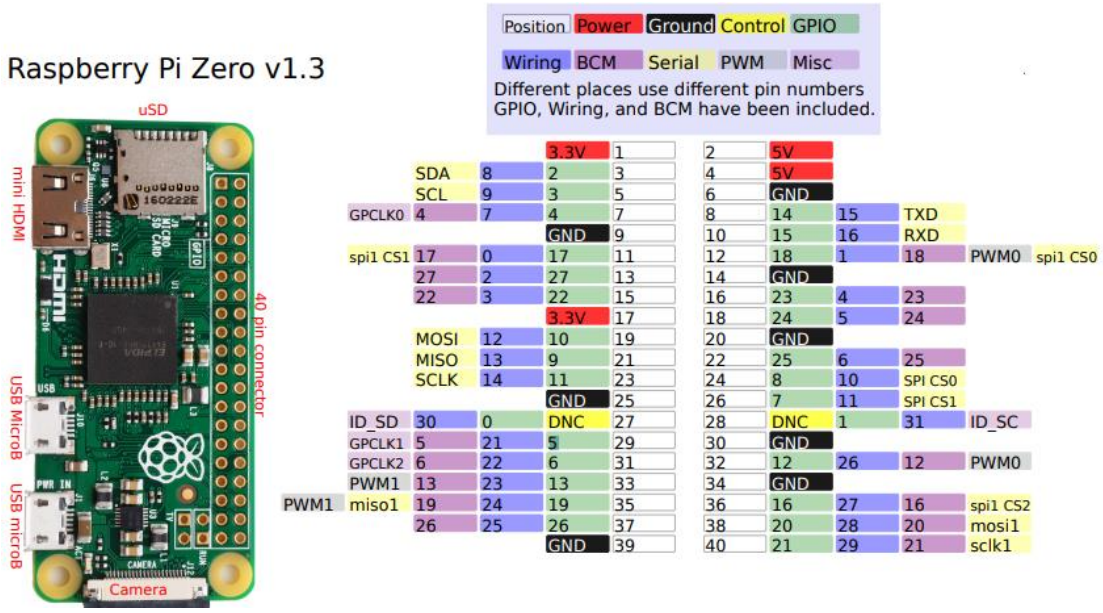

Figura 57. Placa de desarrollo raspberry Pi Zero

Fuente: https://cdn.sparkfun.com/assets/learn_tutorials/6/7/6/PiZero_1.pdf

La Raspberry Pi Zero es un dispositivo en tamaño realmente reducido, teniendo dimensiones de (65x30x5) mm, y el consumo energético que necesita para funcionar es de 350 mA que necesita para funcionar.

Tabla 20. Tabla de Características Raspberry Zero.

Características	
Precio	$45
Procesador	Broadcom BCM2835 single core 1 GHz
Memoria RAM	512 Mb
WLAN	Wi-Fi 802.11b / g / n
Bluetooth	Bluetooth 4.1 BLE
LAN	No dispone
Puertos	2x micro-USB ports (one for power)
Acceso	Extended 40-pin GPIO header
Interfaces para video y sonido	mini HDMI port; CSI camera port
Multimedia	Video composite
Dimensiones	65 x 30 x 5 (mm)
Consumo	5v/350 mA

Fuente: modificado de https://www.raspberrypi.org/products/raspberry-pi-zero/?resellerType=home

- *Raspberry Pi 3 B+*

El Raspberry Pi 3 B + tiene un procesador de cuatro núcleos de 64 bits que funciona a 1.4GHz, LAN inalámbrica de banda dual de 2.4GHz y 5GHz, Bluetooth 4.2 / BLE, Ethernet más rápido y PoE capacidad a través de un PoE HAT separado. La LAN inalámbrica de doble banda viene con certificación de cumplimiento modular, lo que permite que la placa se diseñe en productos finales con pruebas de cumplimiento de LAN inalámbrica significativamente reducidas, lo que mejora tanto el costo como el tiempo de comercialización.

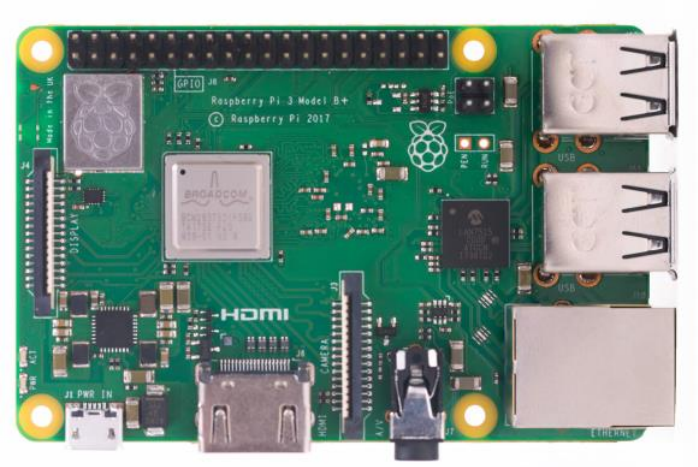

Figura 58. Placa de desarrollo raspberry Pi 3 B+

Fuente: (Pi, n.d.)

Especificaciones:

- CPU: quad core Broadcom BCM2837B0 ARM Cortex-A53 a 1.4GHz el mismo que la Raspberry Pi 3 pero con 200MHz más
- GPU: VideoCore IV GPU
- RAM: 1024MB
- Almacenamiento: ranura microSD
- Vídeo: salida HDMI y vídeo compuesto sin montar el conector
- USB: 4 x USB 2.0
- Red Gigabit limitada a 300Mbps
- Wi-Fi y Bluetooth mejorados con dual band y soporte de 802.11ac wireless LAN y Bluetooth 4.2.
- GPIO: 40 pin
- Alimentación: 2.5A a 5V vía micro USB

- Dimensiones: 86mm x 57mm

- *Raspberry Pi 4*

La Raspberry Pi 4 es uno de los ordenadores más básicos que hoy en día se puede encontrar. Raspberry Pi 4, es una nueva versión con una CPU ARM Cortex-A72 que permite la decodificación de vídeo 4K a 60 fps. A continuación, en la Figura 60 se presenta la placa raspberry pi 4, indicando sus puertos e interfaces.

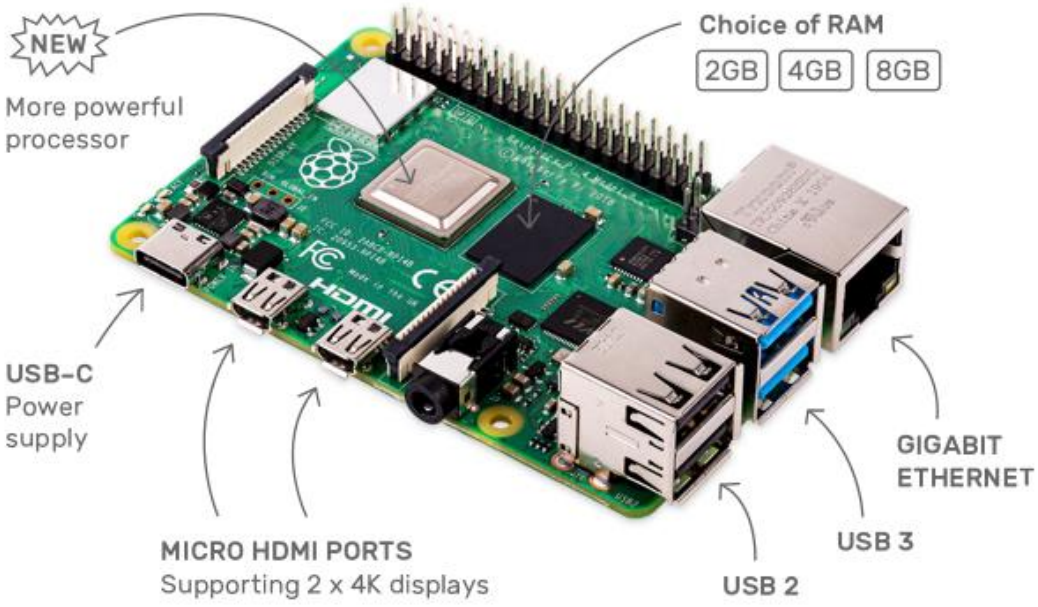

Figura 59. Placa de desarrollo raspberry Pi 4

Fuente: https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica

La raspberry 4 viene con Bluetooth 5.0 y Wi-Fi 802.11ac para las conexiones inalámbricas, de igual manera para su alimentación cuenta con un USB-C que suma 500 mA extra de energía

para alcanzar un total de 1.2 A. A continuación, en la Tabla 16 se presenta las características más importantes.

Tabla 21. Tabla 16 Tabla de Características Raspberry PI 4

RASPBERRY PI 4	
PROCESADOR	ARM Cortex-A72
FRECUENCIA DE RELOJ	1,5 GHz
GPU	VideoCore VI (con soporte para OpenGL ES 3.x)
MEMORIA	1 GB / 2 GB / 4 GB LPDDR4 SDRAM
CONECTIVIDAD	Bluetooth 5.0, Wi-Fi 802.11ac, Gigabit Ethernet
PUERTOS	GPIO 40 pines 2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Micro SD Conector de audio jack USB-C (alimentación)

Fuente: **https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica**

Anexo 7: Creación de etiquetas LabelImg

LabelImg está escrito en Python y usa QT para su interfaz gráfica. Es una forma fácil y gratuita de etiquetar cientos de imágenes para probar su próximo proyecto. LabelImg admite el etiquetado en formato de archivo de texto VOC XML o YOLO (Joseph Nelson, 2020).

Figura 60. Logo Software LabelImg

Fuente: (Joseph Nelson, 2020)

Para iniciar LabelImg en este caso en Windows, no es necesario instalarlo; sólo se debe descargar el programa y tenerlo en una carpeta específica. Una vez descargado el programa, se observa que tenemos un archivo ejecutable y otra carpeta con el nombre de data; damos doble clic en dicha carpeta.

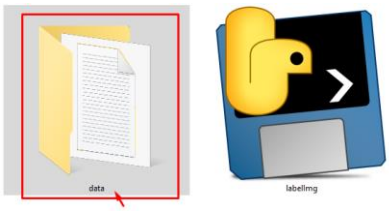

Figura 61. Carpeta de ubicación de la herramienta LabelImg

Fuente*:* Autoría propia

Una vez que ingresemos en la carpeta data, abrimos el archivo txt con el nombre *predefined_classes*; este bloc de notas contiene las etiquetas predefinidas del programa. En este caso se va a dejar solamente con las etiquetas deseadas que en este caso sería: Mipomoea, la cual

debemos anotar y guardar los cambios. Cabe recalcar que es indispensable tomar en cuenta el orden del etiquetado para no tener inconvenientes en el desarrollo del entrenamiento.

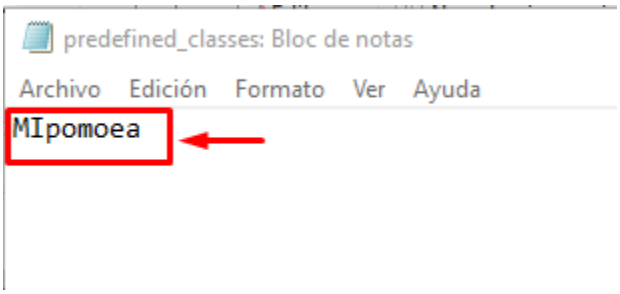

Figura 62. Etiquetas establecidas

Fuente*:* Autoría propia

Definidas las etiquetas, nos dirigimos a la carpeta en donde se encuentra el mismo y damos clic derecho en ejecutar como administrador.

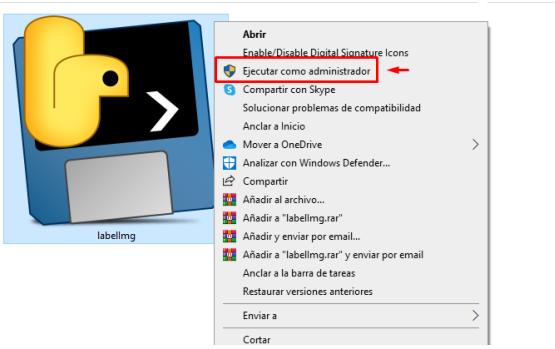

Figura 63. Iniciando LabelImg

Fuente*:* Autoría propia

Una vez abierto el programa; para empezar a etiquetar las imágenes nos dirigimos a la carpeta open, en donde se pone la ubicación exacta de las imágenes.

Figura 64. Herramienta LabelImg

Fuente*:* Autoría propia

Una vez que se tenga la imagen a etiquetar en la plantilla de etiquetado; como primer paso se verifica el formato de la etiqueta, la misma que debe encontrase como YOLO. Seguido del formato, para iniciar a etiquetar las imágenes, se da clic en *Create RectBox*, herramienta que sirve para el cuadro delimitador de la imagen a ser entrenada.

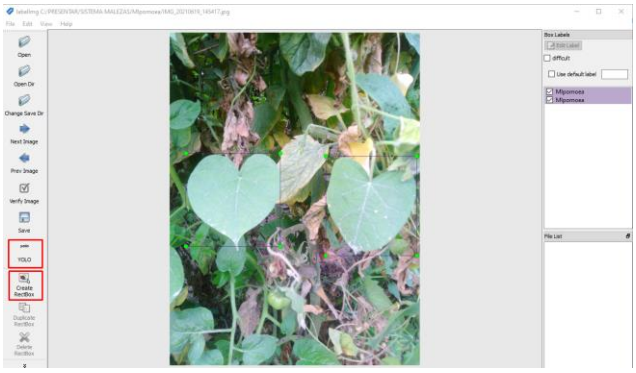

Figura 65. Utilización de la herramienta LabelImg

Fuente*:* Autoría propia

Una vez se tenga la seleccionada la herramienta del etiquetado, se busca el objeto a encerrarse en el cuadro delimitador. Se selecciona el objeto, enseguida aparecen las opciones de

etiquetado, en donde seleccionamos la etiqueta a utilizarse. Se puede verificar que una vez se dé clic en Ok, la etiqueta queda registrada en la parte derecha de los *Box Labels*.

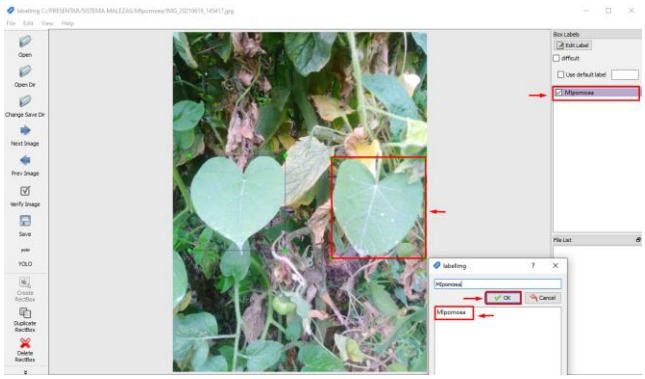

Figura 66. Etiquetamiento de imágenes con la herramienta LabelImg

Fuente*:* Autoría propia

Realizada la etiqueta, se procede a guardarla en la misma ubicación en donde se encuentran las imágenes, la diferencia es que se debe crear una carpeta que contenga los *labels* de cada imagen.

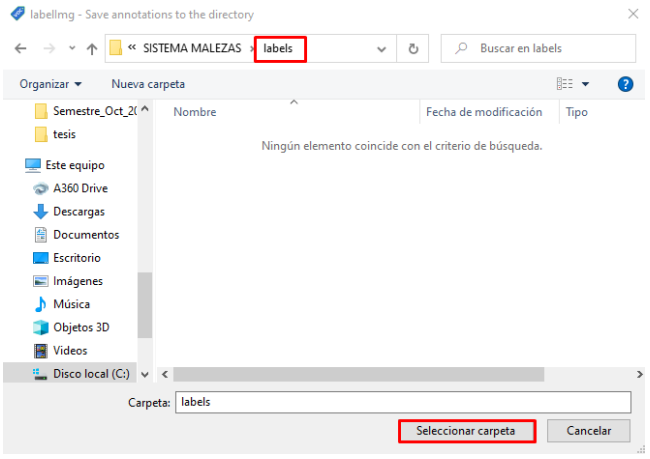

Figura 67. Guardar las imágenes etiquetadas

Fuente*:* Autoría propia

Ya que se ha guardado el etiquetado de la imagen, se verifica en la ubicación seleccionada, que se ha creado un archivo de texto con el nombre de *classes*, el cual contiene las etiquetas creadas. Se puede observar que todos los archivos generados de las etiquetas se guardaron en formato YOLO (.txt). Las mismas que se utilizarán para el entrenamiento del sistema.

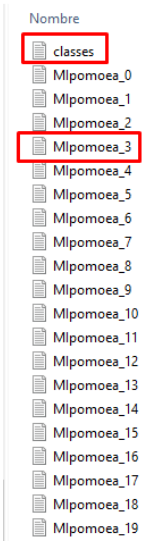

Figura 68: Etiquetas guardadas en formato YOLO

Fuente*:* Autoría propia

Anexo 8: Sistema de reconocimiento de maleza Ipomoea

Dentro del sistema de reconocimiento de la maleza Ipomoea, se puede mencionar que existen archivos dentro del mismo; los cuales se detallan a continuación:

Se tiene la carpeta llamada config_file, en donde se detallan las configuraciones del sistema; dentro de la misma se encuentran los siguientes archivos:

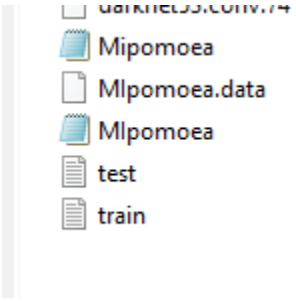

Figura 69. Archivos de configuración

Fuente*:* Autoría propia

- Archivo MIpomoea.cfg: se encuentra todos los archivos de configuración de nuestro sistema; en donde se especifican datos como: el número de filtros, las dimensiones de maxpool, la dimensión de las imágenes a la q se reduce, entre otros.

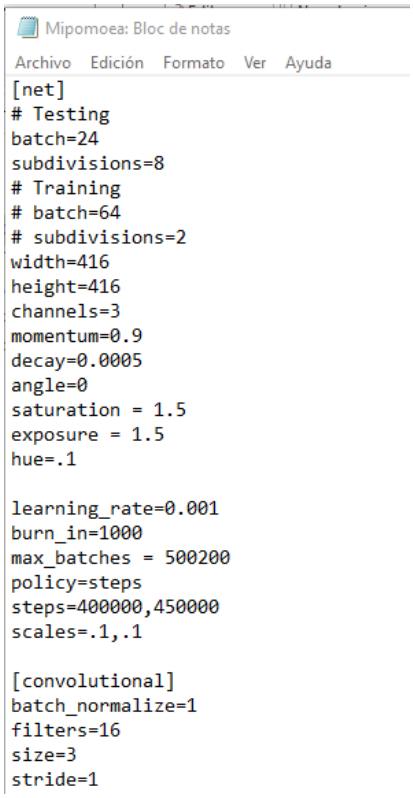

Figura 70. Archivo MIpomoea.cfg

Fuente*:* Autoría propia

- Archivo MIpomoea.names: en este archivo se guarda las etiquetas de salida, es decir el nombre que aparecerá sobre el objeto en el momento del reconocimiento; la etiqueta que se ha puesto es Mipomoea

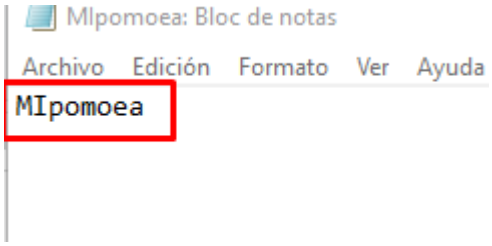

Figura 71. Archivo MIpomoea.names

Fuente*:* Autoría propia

- Archivo MIpomoea.weights: en el cual se tienen todos los pesos correspondientes a cada una de las imágenes entrenadas dentro del sistema

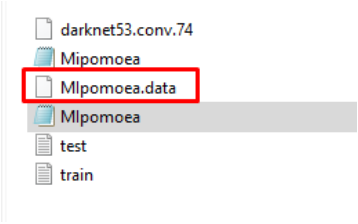

Figura 72. Archivo MIpomoea.weights

Fuente: Autoría propia

Terminada la revisión de la carpeta config_file, se tiene dos carpetas más; una con el nombre img y otra con el nombre video.

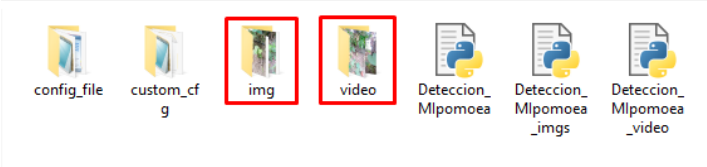

Figura 73. Carpetas para guardar imágenes y videos para el reconocimiento

Fuente*:* Autoría propia

En las cuales se guardará todas las imágenes y los videos a ser reconocidas por sistema respectivamente.

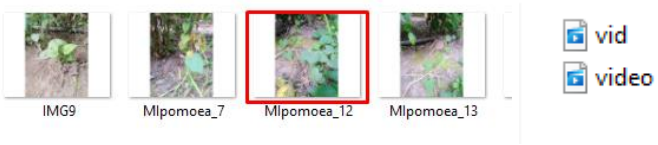

Figura 74. Archivos dentro de las carpetas imagen y video

Fuente*:* Autoría propia

Fuera de esta carpeta, se tiene los códigos correspondientes al reconocimiento de maleza Ipomoea. Se tiene un archivo de reconocimiento en tiempo real, reconocimiento mediante imágenes guardadas (Deteccion_MIpomoea_imgs) y reconocimiento mediante un video (Deteccion_MIpomoea_video); los cuales visualizamos a continuación:

- Reconocimiento en tiempo real

```
#Importamos librerias
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2 as cv
import numpy as np


#Este codigo es parte del desrrollo de trabajo de grado de titulación de la Carrera de Ingeniería en
Electrónica y Redes de Comunicación de la Universidad Técnica del Norte por la Srta. Karla Moncayo

#Dibujamos los cuadros delimitadores de los objetos encontrados
def findObjects(outputs,img):
    hT, wT, cT = img.shape
    bbox = []
    classIds = []
    confs = []
    for output in outputs:
        for det in output:
            scores = det[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                w,h = int(det[2]*wT) , int(det[3]*hT)
                x,y = int((det[0]*wT)-w/2) , int((det[1]*hT)-h/2)
                bbox.append([x,y,w,h])
                classIds.append(classId)
                confs.append(float(confidence))

    indices = cv.dnn.NMSBoxes(bbox, confs, confThreshold, nmsThreshold)

    for i in indices:
        i = i[0]
        box = bbox[i]
        x, y, w, h = box[0], box[1], box[2], box[3]
        # print(x,y,w,h)
        cv.rectangle(img, (x, y), (x+w,y+h), (255, 0 , 0), 2)
        cv.putText(img,f'{classNames[classIds[i]].upper()} {int(confs[i]*100)}%',
                   (x, y-10), cv.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
```

Figura 75. Código para reconocimiento en tiempo real

Fuente*:* Autoría propia

- Reconocimiento mediante imágenes guardadas

```
import cv2 as cv
import numpy as np

#Este codigo es parte del desrrollo de trabajo de grado de titulación de la Carrera de Ingeniería en
Electrónica y Redes de Comunicación de la Universidad Técnica del Norte por la Srta. Karla Moncayo

def findObjects(outputs,img):
    hT, wT, cT = img.shape
    bbox = []
    classIds = []
    confs = []
    for output in outputs:
        for det in output:
            scores = det[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                w,h = int(det[2]*wT) , int(det[3]*hT)
                x,y = int((det[0]*wT)-w/2) , int((det[1]*hT)-h/2)
                bbox.append([x,y,w,h])
                classIds.append(classId)
                confs.append(float(confidence))

    indices = cv.dnn.NMSBoxes(bbox, confs, confThreshold, nmsThreshold)

    for i in indices:
        i = i[0]
        box = bbox[i]
        x, y, w, h = box[0], box[1], box[2], box[3]
        # print(x,y,w,h)
        cv.rectangle(img, (x, y), (x+w,y+h), (255, 0 , 0), 2)
        cv.putText(img,f'{classNames[classIds[i]].upper()} {int(confs[i]*100)}%',
                   (x, y-10), cv.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
```

Figura 76. Código de reconocimiento mediante imágenes guardadas

Fuente*:* Autoría propia

- Reconocimiento mediante un video

```
#Importamos librerias
import cv2 as cv
import numpy as np

 #Este codigo es parte del desrrollo de trabajo de grado de titulación de la Carrera de Ingeniería en
Electrónica y Redes de Comunicación de la Universidad Técnica del Norte por la Srta. Karla Moncayo

#funcion para dibujar los objetos encontrados
def findObjects(outputs,img):
    hT, wT, cT = img.shape
    bbox = []
    classIds = []
    confs = []
    for output in outputs:
        for det in output:
            scores = det[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                w,h = int(det[2]*wT) , int(det[3]*hT)
                x,y = int((det[0]*wT)-w/2) , int((det[1]*hT)-h/2)
                bbox.append([x,y,w,h])
                classIds.append(classId)
                confs.append(float(confidence))

    indices = cv.dnn.NMSBoxes(bbox, confs, confThreshold, nmsThreshold)

    for i in indices:
        i = i[0]
        box = bbox[i]
        x, y, w, h = box[0], box[1], box[2], box[3]

        cv.rectangle(img, (x, y), (x+w,y+h), (255, 0 , 0), 2)
        cv.putText(img,f'{classNames[classIds[i]].upper()} {int(confs[i]*100)}%',
                  (x, y-10), cv.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
```

Figura 77. Código de reconocimiento mediante videos guardados

Fuente*:* Autoría propia

BIBLIOGAFÍA

AgroJornada. (2018). *Tipos de Malezas en los Cultivos: ¿Cuáles Hay y Cómo Combatirlas?* https://agrojornada.com.py/tipos-de-malezas-en-los-cultivos/

Aguirre, Jaramillo, & Quizhpe. (2019). *Arvenses asociadas a cultivos y pastizales del Ecuador*. www.ediloja.com.ec

Alvarado, C. (2018). *Clasificación de tweets mediante modelos de aprendizaje supervisado*.

Andrea, C. C., Mauricio Daniel, B., & Jose Misael, J. B. (2018). Precise weed and maize classification through convolutional neuronal networks. *2017 IEEE 2nd Ecuador Technical Chapters Meeting, ETCM 2017*, *2017-Janua*, 1–6. https://doi.org/10.1109/ETCM.2017.8247469

ANDRÉS MONTENEGRO, C. P. (2015). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE MALEZAS EN CULTIVOS CUNDIBOYACENSES. 数据采集与处理, *1*(30), 77–87.

Antonio. (2016). *Características de la planta de tomate (con FOTOS)*. https://www.mundohuerto.com/cultivos/tomate/caracteristicas

Bariga, E. R. C. (2017). APLICACIÓN PRÁCTICA DE LA VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE ROSTROS EN UNA IMAGEN, UTILIZANDO REDES NEURONALES Y ALGORITMOS DE RECONOCIMIENTO DE OBJETOS DE LA BIBLIOTECA OPENCV. *Chemosphere*, *7*(1), 13–19. https://doi.org/10.1016/j.jenvman.2018.01.013

Bennett, D. M. (2014). [No Title]. *British Journal of Psychiatry*, *205*(01), 76–77. https://doi.org/10.1192/bjp.205.1.76a

Bootcamp AI. (2019, November 23). *Intro a las redes neuronales convolucionales | by Bootcamp AI | Medium*. https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8

Calderón, L. E. (2012). Clase, UNAM 2006, Fisica, Digitalizacion imagenes. *Procesamiento Digital de Imagenes*, 1–7. https://www.google.com.mx/#sclient=psy-ab&hl=es-419&site=&source=hp&q=Calderón+Aguilera+LE+(2006)+A+la+Cima+de+Baja+California&oq=Calderón+Aguilera+LE+(2006)+A+la+Cima+de+Baja+California&gs_l=hp.3...641.641.1.1438.1.1.0.0.0.0.78.78.1.1.0...0.0...1c.1.mA3k

Calvo, D. (2018). *Función de activación - Redes neuronales - Diego Calvo*. http://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/

Calvo, D. (2019). *Aprendizaje supervisado - Diego Calvo*. http://www.diegocalvo.es/aprendizaje-supervisado/

Enrique. (2018). *Detección de objetos con YOLO: implementaciones y como usarlas*. https://medium.com/@enriqueav/detección-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246

Flores. (2014). *Maleza | Qué es, características, tipos, porqué es indeseable | Planta*. https://www.flores.ninja/maleza/

Galán, H., & Martínez, A. (2015). Inteligencia artificial . Redes neuronales y Aplicaciones. *Universidad Carlos III de Madrid, Journal*, 8. http://www.it.uc3m.es/jvillena/irc/practicas/10-11/06mem.pdf

García. (2020). *Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles*.

García, I. (2015). *La visión artificial y los campos de aplicación*. *1*, 94–103.

Godoy, B. S. N. (2016). *Universidad Técnica Del Norte Facultad De Ingeniería Y Ciencias Aplicadas*.

Gómez, J. (2015). *A 18 de Febrero de 2015* . 2015. https://doi.org/10.1007/s12033-014-9823-4.-

González, L. (2018). *Todo sobre aprendizaje supervisado en Machine Learning - Ligdi González*. http://ligdigonzalez.com/todo-sobre-aprendizaje-supervisado-en-machine-learning/

Guti, C. (2019). *E . T . S . de Ingeniería Industrial , Informática y de Telecomunicación Detección de armas en vídeos mediante técnicas de Deep Learning Grado en Ingeniería Informática Trabajo Fin de Grado Resumen*. 1–60.

Infaimon. (2018). *Visión por computador: qué es y cuáles son sus usos más comunes*. https://blog.infaimon.com/vision-computador-soluciones-permite/

ISO/IEC/IEEE 29148. (2011). *ISO - ISO/IEC/IEEE 29148:2011 - Systems and software engineering — Life cycle processes — Requirements engineering*. https://www.iso.org/standard/45171.html

Jhan, G., & Restrepo Arteaga, P. (2015). *APLICACIÓN DEL APRENDIZAJE PROFUNDO (" DEEP LEARNING ") AL PROCESAMIENTO DE SEÑALES DIGITALES*. https://red.uao.edu.co/bitstream/10614/7975/1/T05978.pdf

Joseph Nelson. (2020, March 16). *Getting Started with LabelImg for Labeling Object Detection Data*. https://blog.roboflow.com/labelimg/

Magap. (2017). *Cifras Agroproductivas*. http://sipa.agricultura.gob.ec/index.php/cifras-agroproductivas

Massiris Manlio , Juan Álvaro Fernández Muñoz, C. D. (2018). *Detección De Equipos De Protección Personal Mediante Red Neuronal Convolucional Yolo*. *September*. https://www.researchgate.net/publication/327449170

Monsanto. (2017). *¿Qué es la maleza y por qué es mala?* https://www.hablemosdelcampo.com/que-es-la-maleza-y-por-que-es-mala/

Nesheim, M. C., Oria, M., Yih, P. T., & Board, N. (2015). *A Framework for Assessing Effects of the Food System*. https://doi.org/10.17226/18846

Ortega, E. E. (2015). *Manejo integrado de malezas en el cultivo de tomate*.

Pablo, & Toro, G. & M. O. DEL. (2013). *Univer rsidad Po litécnica d Valenc de*.

Pardo, E., & Saelzer, P. (2006). *Universidad Nacional Agraria Facultad De Ciencia Animal Obstetricia Y Ginecologia*.

Patricia, N., & Arredondo, A. (2009). *Método Semisupervisado para la Clasificación Automática de Textos de Opinión*.

Pérez, L. S. M. (2012). *InGeNiEria De SofTwArE: MODELO EN V*. http://softwareverde.blogspot.com/2012/09/modelo-en-v.html

Pi, R. (n.d.). *Raspberry Pi 3 Model B +*.

Pichisaca, M. (2003). *El cultivo de Tomate Riñón en Invernadero (Lycopersicon esculentum) Cultivo de tomate riñón en invernadero*. 59. https://digitalrepository.unm.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1366&context=abya_yala

Pineda, P. (2018). *Diseño De Un Sistema De Reconocimiento Automático De Vehículos Mediante El Uso De Redes Neuronales Profundas (Dnn)*.

Productor, E. (2018). *MAG capacita en cultivo de tomate riñón bajo invernadero | Noticias Agropecuarias*. https://elproductor.com/noticias/mag-capacita-en-cultivo-de-tomate-rinon-bajo-invernadero/

Quintero, C. (2018). Uso de Redes Neuronales Convolucionales para el Reconocimiento

Automático de Imágenes de Macroinvertebrados para el Biomonitoreo Participativo. *KnE Engineering*, *3*(1), 585. https://doi.org/10.18502/keg.v3i1.1462

RNeuronales. (2012). *Elementos Básicos de las Redes Neuronales | rneuronales*. https://rneuronales.wordpress.com/2012/11/07/elementos-basicos-de-las-redes-neuronales/

Román, V. (2019). *Aprendizaje No Supervisado en Machine Learning: Agrupación*. https://medium.com/datos-y-ciencia/aprendizaje-no-supervisado-en-machine-learning-agrupación-bb8f25813edc

Santos, P. R. de los S. (2017). *Los 2 tipos de aprendizaje en Machine Learning: supervisado y no supervisado - Think Big Empresas*. https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/

Soler Martínez Claudia. (2016). *ASPECTOS FISIOLOGICOS Y MORFOLOGICOS DE LAS MALEZAS PROF. PEDRO RODRIGUEZ ESPECIALISTA EN HERBOLOGIA - PDF Free Download*. https://docplayer.es/5934514-Aspectos-fisiologicos-y-morfologicos-de-las-malezas-prof-pedro-rodriguez-especialista-en-herbologia.html

Terán, C. A. (2020). *¿Técnicas de recolección de datos para realizar un trabajo de investigación? | Online Tesis*. https://online-tesis.com/tecnicas-de-recoleccion-de-datos-para-realizar-un-trabajo-de-investigacion/

Uribe, S. E. (2018). *Implementación de redes neuronales y análisis de rendimiento para sistemas empotrados*.

UTN. (2019). *Misión y Visión*. https://www.utn.edu.ec/fica/carreras/electronica/?page_id=9

Vallejo, S. (2016). *Modelo V / Ciclo de vida del software*. http://cdvmv.blogspot.com/2016/02/modelo-v.html

Vega, H., Cortez, A., Huayna, A., Loayza, L. y Naupari, P. (2009). Reconocimiento de patrones

mediante redes neuronales artificiales. *Revista de Ingeniería de Sistemas e Informática*, *6*(2), 17–26. http://sisbib.unmsm.edu.pe/Bibvirtual/publicaciones/risi/2009_n2/v6n2/a03v6n2.pdf

Yujato, W. X. G. (2016). *UNIVERSIDAD CENTRAL DEL ECUADOR " IDENTIFICACIÓN DE ARVENSES PRESENTES EN EL CULTIVO DE CACAO (Theobroma cacao L .) EN MILAGRO , NARANJAL Y NARANJITO " TESIS DE GRADO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERA DORIS TATIANA PILLAJO LUGUAÑA QUITO-ECUA*.

Zambrano, A. P. (2009). *Cultivo de tomate en invernadero*.

Zhang, W., Hansen, M. F., Volonakis, T. N., Smith, M., Smith, L., Wilson, J., Ralston, G., Broadbent, L., & Wright, G. (2018). Broad-Leaf Weed Detection in Pasture. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 101–105. https://doi.org/10.1109/ICIVC.2018.8492831