UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSGRADO

MAESTRÍA EN TELECOMUNICACIONES

“RECONOCIMIENTO DE LESIONES NECRÓTICAS PARA LA DETECCIÓN TEMPRANA DE LA PLAGA THRIPS (KAKOTHRIPS ROBUSTUS UZEL) EN LOS CULTIVOS DEL GUISANTE O ARVEJA MEDIANTE TÉCNICAS DEEP LEARNING”

Trabajo de Investigación previo a la obtención del Título de Magíster en Telecomunicaciones.

AUTOR(A):

CARLOS JONATHAN GUERRERO ANDRADE

DIRECTOR(A):

ING. SILVIA DIANA MARTÍNEZ MOSQUERA MSC

IBARRA - ECUADOR

2021

APROBACIÓN DEL TUTOR

Yo, **Silvia Diana Martínez Mosquera**, certifico que el estudiante **Carlos Jonathan Guerrero Andrade** con Cédula **N°040105127-1** ha elaborado bajo mi tutoría la sustentación del trabajo de grado titulado: **"Reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips (Kakothrips Robustus Uzel) en los cultivos del guisante o arveja mediante técnicas de Deep Learning."**

Este trabajo se sujeta a las normas y metodologías dispuestas en el reglamento del título a obtener, por lo tanto, autorizo la presentación a la sustentación para la calificación respectiva.

Ibarra, 21 de octubre del 2021

MSc. Silvia Diana Martínez Mosquera

Tutor

C.I.: 1718478603

UNIVERSIDAD TÉCNICA DEL NORTE
INSTITUTO DE POSTGRADO
BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

En el 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 de la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	0401051271
APELLIDOS Y NOMBRES:	Guerrero Andrade Carlos Jonathan
DIRECCIÓN:	Ciudadela del Chofer Chile 6-23 y Bolivia
EMAIL:	cjguerreroa@utn.edu.ec
TELÉFONO MÓVIL:	0967116077
DATOS DE LA OBRA	
TÍTULO:	"Reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips (Kakothrips Robustus Uzel) en los cultivos del guisante o arveja mediante técnicas Deep Learning"
AUTOR:	Guerrero Andrade Carlos Jonathan
FECHA:	21 de octubre del 2021
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	( ) PREGRADO (X) POSGRADO
TÍTULO POR EL QUE OPTA:	Magister en Telecomunicaciones
ASESOR/DIRECTOR:	MSc. Silvia Diana Martínez Mosquera

2. CONSTANCIA

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

Ibarra, a los 21 del mes de octubre del 2021

Carlos Jonathan Guerrero Andrade

C.I.:0401051271

DEDICATORIA

El presente trabajo investigativo está dedicado. A mi madre, por ser el pilar más importante en mi vida y demostrarme su amor y apoyo incondicional en todo momento. A mi padre y hermanos, que con sus palabras de aliento, comprensión, cariño y respeto me han permitido cumplir con un anhelo más para mi formación profesional y de ser humano.

RECONOCIMIENTO

A Dios, por ser mi guía en el transcurso de mi existencia y la fuente inspiradora, que me ha brindado sabiduría y paciencia para culminar una etapa más en mi vida.

A la Ing. Diana Martínez MSc., por la tutoría de este trabajo de grado quien, con su conocimiento, sabiduría y experiencia profesional, supo guiarme en todo el proceso de investigación y sobre todo a la paciencia que me tuvo para que este trabajo llegue a su fin.

Al Ing. Marco Yaselga MSc., por el asesoramiento, consejos y enseñanzas que me brindo, los que permitieron enaltecer el desarrollo de la presente investigación.

A todos los docentes por haber compartido su sabiduría y experiencias invaluables a lo largo del programa de maestría. A mis compañeros de cohorte por los buenos momentos que compartimos, sobre todos aquellos que de alguna manera aportaron con sus conocimientos en alguna área específica los mismos que supieron enriquecernos a todos.

De manera especial a la Sra. Maritza Flores, por sus consejos y palabras de aliento que me supo brindar en todo momento.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

UNIVERSIDAD TÉCNICA DEL NORTE
INSTITUTO DE POSGRADO
PROGRAMA DE MAESTRÍA

"RECONOCIMIENTO DE LESIONES NECRÓTICAS PARA LA DETECCIÓN TEMPRANA DE LA PLAGA THRIPS (KAKOTHRIPS ROBUSTUS UZEL) EN LOS CULTIVOS DEL GUISANTE O ARVEJA MEDIANTE TÉCNICAS DEEP LEARNING"

Autor: Ing. Carlos Jonathan Guerrero Andrade
Tutor: Ing. Silvia Diana Martínez Mosquera MSc.
Año: 2021

RESUMEN

En la actualidad, el seguimiento de cultivos en las parcelas agrícolas sigue siendo una de las tareas más trascendentales que tiene agricultura de precisión, ya que por medio esta se puede efectuar la estimación del rendimiento y la predicción de cosechas. Debido a las complicadas condiciones atmosféricas y factores climáticos que presenta el sector, la detección temprana de plagas y enfermedades se ha convertido en un desafío considerable que los productores deben asumir de manera constante. Esta investigación propone un sistema de reconocimiento rápido y eficaz de lesiones necróticas para la detección temprana de la plaga thrips en el guisante o arveja mediante la implementación de técnicas de *Deep Learning*. Para el sistema se utilizaron: un Smartphone y el drone DJI Mavic mini, para recopilar imágenes JPG (*Joint Photographic Experts Group*) desde las parcelas del cultivo. A continuación, se utilizó la herramienta Roboflow para la extracción, etiquetado, segmentación de las características significativas y relevantes del objeto de estudio en cada una de las imágenes. Fue propuesta la arquitectura de la red neuronal convolucional *Yolov4-Tiny*, para la detección de la plaga thrips en los cultivos. En el desarrollo del sistema se emplearon herramientas de software libre, como *Python* y librerías de aprendizaje automático como *Tensorflow* y de visión artificial como *OpenCV*. Finalmente, el sistema desarrollado fue puesto a las respectivas pruebas de funcionamiento. Donde los resultados experimentales mostraron que la IoU (Intersección sobre la Unión) es de 59,23% para una mAP (Precisión Media) de 87,8% sobre un conjunto de datos de alta densidad. Los resultados alcanzados por el sistema fueron comparados con el diagnóstico previo realizado el experto en producción del cultivo a través de observaciones directas al guisante o arveja. Donde se pudo a establecer que el sistema tiene una efectividad del 80%.

Palabras claves: *Plaga Thrips, Deep Learning, Yolov4-Tiny, TensorFlow, OpenCV.*

UNIVERSIDAD TÉCNICA DEL NORTE
INSTITUTO DE POSGRADO
PROGRAMA DE MEASTRÍA

"RECONOCIMIENTO DE LESIONES NECRÓTICAS PARA LA DETECCIÓN TEMPRANA DE LA PLAGA THRIPS (KAKOTHRIPS ROBUSTUS UZEL) EN LOS CULTIVOS DEL GUISANTE O ARVEJA MEDIANTE TÉCNICAS DEEP LEARNING"

Autor: Ing. Carlos Jonathan Guerrero Andrade
Tutor: Ing. Silvia Diana Martínez Mosquera MSc.
Año: 2021

ABSTRACT

At present, the monitoring of crops in agricultural plots remains one of the most important tasks that precision agriculture has, since through it is possible to perform the estimation of the yield and the prediction of crops. Due to the complicated atmospheric conditions and climatic factors that the sector presents, they have made the early detection of pests and diseases a considerable challenge that producers must constantly assume. This research proposes a rapid and effective recognition system for necrotic lesions for the early detection of the thrips infestation in peas or peas through the implementation of Deep Learning techniques. For the system, a smartphone and the DJI Mavic mini were used to collect JPG (Joint Photographic Experts Group) images of the growing plots. Next, the Roboflow tool was used for the extraction, labeling and segmentation of the significant and relevant characteristics of the study object in each of the images. The Yolov4-Tiny convolutional neural network architecture was proposed for the detection of thrips infestation in crops. In the development of the system, free software tools such as Python and machine learning libraries such as Tensorflow and artificial vision libraries such as OpenCV were used. Finally, the developed system was subjected to the respective functional tests. Where the experimental results showed that the IoU (Intersection over the Union) is 59.23% for a mAP (Medium Precision) of 87.8% in a high-density data set. The results obtained by the system were compared with the previous diagnosis made by the expert in crop production through direct observations of the pea or pea. Where it could be established that the system is 80% effective.

Keywords: *Plaga Thrips, Deep Learning, Yolov4-Tiny, TensorFlow, OpenCV.*

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

Tema:

"RECONOCIMIENTO DE LESIONES NECRÓTICAS PARA LA DETECCIÓN TEMPRANA DE LA PLAGA THRIPS (KAKOTHRIPS ROBUSTUS UZEL) EN LOS CULTIVOS DEL GUISANTE O ARVEJA MEDIANTE TÉCNICAS DEEP LEARNING"

Problema de la Investigación

Contextualización del problema

La Comisión Económica para América Latina y el Caribe (CEPAL), así como la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) y el Instituto Interamericano de Cooperación para la Agricultura (IICA). En un informe referente a los años 2019-2020, mencionan que no se puede reducir la pobreza, la malnutrición y el cambio climático; si la sociedad y los sectores políticos no proyectan al sector rural como motor de desarrollo económico, social y ambiental. Además, señalan que la agricultura debe ser considerada como parte fundamental del impulso económico, para que promueva el desarrollo sostenible en los territorios rurales (CEPAL, 2020).

Actualmente, la agricultura es un factor importante para la economía y la sociedad del mundo; y según apreciaciones de la FAO, muestra que el potencial agrícola de la LAC (América Latina y el Caribe) en los años posteriores podría sobrepasar al que tiene actualmente los Estados Unidos. Debido a que en los últimos años la LAC, ha sido uno

de los principales actores en el abastecimiento de alimentos a nivel mundial. Pero a inicio de los 90 se pudo evidenciar variaciones en los resultados, esto se debe en gran parte a los factores climáticos. Este tipo de circunstancias en las últimas décadas, han ocasionado pérdidas en el sector agrícola y pecuario de 13.000 millones de dólares, y solo en el cultivo de leguminosas las pérdidas alcanzaron los 8.000 millones de dólares entre los años 2005 y 2015 (CGSpace, 2018).

En Ecuador, el Plan Nacional de Desarrollo (PND), en relación al eje de Economía al Servicio de la Sociedad se menciona que: en los últimos años se han realizado avances en el sector agrícola, pero aún existe baja productividad en los productos básicos y en los de exportación. Señalando que, todavía existe el desafío para trabajar en la democratización de los medios de producción de las distintas unidades productivas, mediante el fortalecimiento de programas de asistencia técnica, así como la capacitación e innovación en el sector agrícola (Secretaría Técnica Planificación Ecuador, 2017).

Las condiciones ambientales y los factores climáticos, en los últimos años juegan un papel importante para la presencia de plagas y enfermedades en la agricultura. Siendo los thri ps[1] una de las principales plagas que afectan a la mayoría de cultivos, provocando en estos diferentes síntomas. Por ejemplo, en el pepino los thrips causan deformación en los frutos, mientras que en el pimiento ocasionan daños estéticos. En cambio, en algunas plantas ornamentales provocan deformación y decoloración a las flores, hojas y tallos. Un pequeño grupo de este tipo de plaga, puede ser más que suficiente para ocasionar daños

[1] Thrips: son pequeños patógenos que afectan a los cultivos mediante la extracción de fluidos de las células vegetales, las misma que al estar vacias se llenan de aire causando que los tallos, hojas y futos cambian el aspecto y color.

irreparables a los cultivos, además son considerados como importantes portadores para la transmisión de virus (Prado Jiménez, 2019).

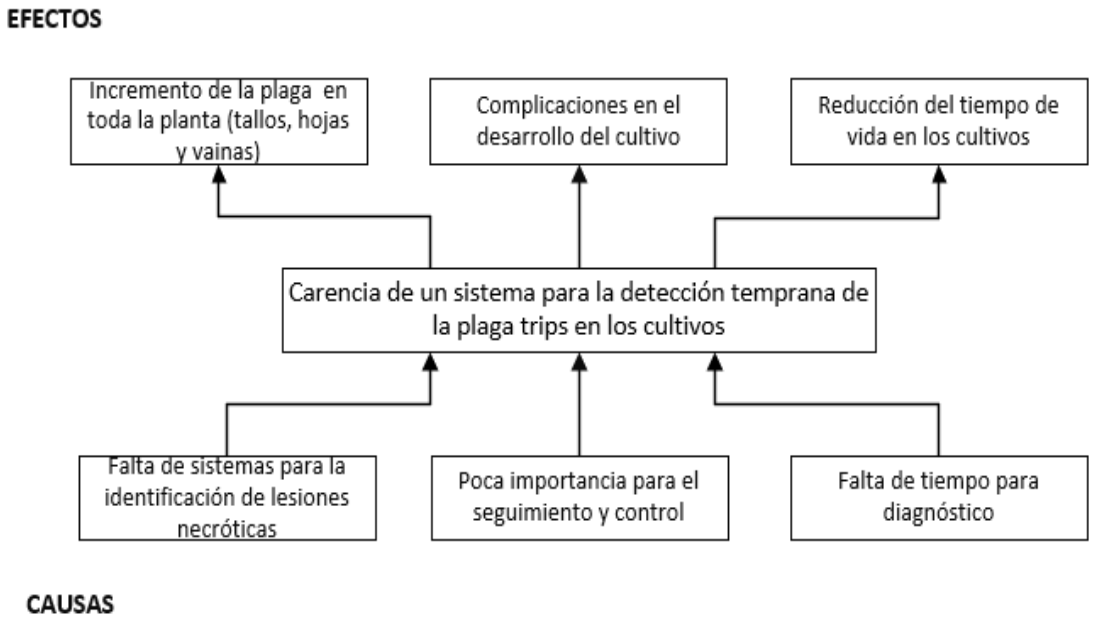
Debido al incremento de plagas en los cultivos, algunos sectores como las entidades gubernamentales y la industria agrícola, han venido utilizando plaguicidas y pesticidas como parte integral de la agricultura. El uso constante de estos insumos, han permitido de alguna forma reducir las pérdidas de los cultivos de un 35 %; cuando son aplicados de manera directa. Aunque, los plaguicidas y pesticidas son todavía útiles en la agricultura moderna, estos siguen siendo sumamente peligrosos para los seres humanos y el medio ambiente (Bereinsten & Edan, 2017).

Análisis crítico

En la Figura 1, se muestra el problema identificado que es la carencia de sistema de reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips (Kakothrips Robustus Uzel) en el guisante o arveja. A pesar de no contar, con un sistema para el reconocimiento del estado actual de los cultivos, las detecciones se han venido realizando de manera artesanal, a través del seguimiento y observaciones directas al guisante o arveja.

Figura 1

Diagrama del árbol de problema de la investigación



Nota: Representación del árbol del problema de la investigación con sus causas y efectos. Realizado por el Autor.

Entre las causas y los efectos del problema identificado están los siguientes:

- Falta de sistemas informáticos para el reconocimiento automático de lesiones necróticas en los cultivos, cuya consecuencia es el incremento de las mismas en todas las plantas.
- Poca importancia en el seguimiento y control de los cultivos, trae como consecuencia el avance de la plaga en el guisante, cuyo efecto ocasiona complicaciones en su desarrollo.
- Falta de tiempo para un diagnóstico temprano tiene como consecuencia la reducción del tiempo de vida del cultivo y pérdida total del mismo.

Planteamiento del Problema

De acuerdo a un reporte del MAGAP (2019), señala que: en el Ecuador el sector agrícola aportó con el 8% a la producción anual de la nación, siendo esta la actividad que más produce empleos. Sin embargo, debido a los grandes avances tecnológicos que actualmente son utilizados en el sector agrícola, la producción en los campos sigue siendo todavía muy baja en relación a otros países (Villota Neira, 2019).

Por ejemplo, la producción de leguminosas como el guisante o arveja, en el 2017 disminuyó en un 39% respecto al año 2016. De igual manera, la producción del guisante o arveja seca presentó una reducción de 32% respecto al año 2016 alcanzando 702 toneladas. Este comportamiento está relacionado con la reducción simultánea de la superficie de cosechas, los factores climáticos, la calidad del cultivo y el rendimiento del producto (MAGAP, 2018).

La provincia del Carchi, en la última década ha sido una de las zonas donde ha existido mayor producción del guisante o arveja. El cultivo de esta leguminosa, ha presentado dificultades muchas de estas se deben a algunos factores climáticos como son las fuertes heladas y las largas sequías. De igual manera, la presencia de problemas fitosanitarios en la etapa de germinación y desarrollo del cultivo han causado problemas en la producción del mismo (Delgado Chamorro, 2014).

Los anteriores inconvenientes, acompañados de las constantes precipitaciones en el sector y la alta humedad que tienen los terrenos han provocado en algunas ocasiones la pérdida total del guisante o arveja. Este tipo de circunstancias de alguna u otra manera

crean las condiciones más adecuadas para que exista la presencia de plagas y enfermedades en los cultivos (Delgado Chamorro, 2014).

Una de las principales plagas que se presenta con frecuencia en el guisante o arveja son los famosos thrips (Kakothrips Robustus Uzel), este es un patógeno que provoca lesiones necróticas en las vainas, hojas y tallos. Además, provoca la decoloración en los cultivos debido a las picaduras que causan en las diferentes partes de la planta. La presencia de este tipo de lesiones en el guisante o arveja es de gran importancia, puesto que ocasionan un impacto productivo y económico (Delgado Chamorro, 2014).

En el guisante o arveja la plaga thrips, pueden reducir las cosechas entre 20 y 50% deteriorando la calidad de la vaina y del grano cosechado (Sánchez Arteaga, 2019). La mala calidad y baja producción se debe a que el sector agrícola ha venido trabajando bajo un sistema tradicional. Debido a esta situación los productores, se ven obligados a utilizar plaguicidas y pesticidas de forma continua sin medir los riesgos que causan y los impactos que provocan al medio ambiente (Toledo Perdomo & Sagastume Mena, 2018).

Para reducir los impactos ambientales y solventar inconvenientes causados por la agricultura tradicional, la aplicación de la tecnificación en los cultivos ha permitido optimizar recursos, mejorar la productividad y reducir la presencia de plagas o enfermedades. Además, que permite intervenir de forma directa y específica en los cultivos a través del seguimiento y monitoreo en tiempo real de parámetros como humedad, temperatura, gases y estado de los cultivos (Patil & Sachapara, 2018).

Formulación del Problema

La falta de un sistema de reconocimiento de lesiones necróticas para el guisante o arveja, hace que no se pueda detectar tempranamente la presencia de la plaga thrips en las plantas, ocasionado la pérdida del cultivo.

Justificación de la Investigación

En el Ecuador, el guisante o arveja es una de las leguminosas más consumidas después del fréjol, debido a que es considerado como uno de los productos más importantes en la dieta de las personas. Este tipo de leguminosa es cultivado en todas las provincias de la región andina del país, siendo la provincia del Carchi la que mayor producción registra con un aproximado de 9,462 hectáreas por año, y con un rendimiento promedio de 8 toneladas por hectárea que representa el 47,46% de la producción nacional (Angulo Pérez, 2019).

Según (Tirira Tirira , 2018), menciona que las variedades de mayor producción de guisante o arveja en el Carchi son la Quantum, la cual se cultiva entre los 2.600 hasta 2.900 m.s.n.m. Los cantones que más cultivan este producto son Bolívar, Espejo, Montúfar. Mientras que la Obonuco andina, es cultivada en zonas de 2.700 a 3.300 m.s.n.m., y su producción se sitúa en los cantones Huaca, Tulcán y Montúfar.

En la Tabla 1, se muestra la producción del guisante o arveja en la provincia del Carchi.

Tabla 1

Producción del guisante o arveja en la provincia del Carchi.

Producto	Cantón	Extensión en (Has)	Porcentaje %	Producción (TM)
Arveja (Pisum Sativum**)**	Bolívar	957	45,68	2.730,29
	Espejo	254	12,12	724,41
	Mira	224	10,72	640,73
	Montúfar	603	28,80	1.721,38
	Huaca	32	1,54	92,05
	Tulcán	24	1,14	68,14
Total		2.094	100,00	5.977,00

Nota: Producción del guisante o arveja en la provincia del Carchi. Tomado *de Estudio de factibilidad de un centro de acopio para la comercialización de arveja tierna (pisum sativum) en la provincia del Carchi* [Tabla], Tirira Tirira , 2018, www. repositorio.utn.edu.ec.

Desde los años 90, las malas prácticas agrícolas, así como la falta de agua y la aplicación constante de fertilizantes, habrían contribuido a que exista una degradación en la producción de cultivos en un 38%. Estos factores contribuidos a las malas condiciones climáticas y el incremento elevado de los costos en insumos y fertilizantes, han hecho posible el uso de la tecnificación en la agricultura (Vasudevan, Kumar, & Bhuvaneswari, 2016).

La implementación de nuevas tendencias tecnológicas como la visión artificial y el aprendizaje automático son cada vez más utilizados, para el análisis de los cultivos. Debido a que, ayuda a realizar de manera remota tareas como: la observación y el procesamiento de imágenes, la identificación de plagas, la medición del estrés en las plantas y la revisión del estado actual de los cultivos. Además, permite efectuar de forma automática el control, planificación y predicción de las cosechas.

La presente investigación se enmarca en el quinto objetivo, del Plan Nacional de Desarrollo (2017), que hace referencia al impulso la productividad y competitividad para el crecimiento económico sostenible de manera redistribuida y solidaria. Actualmente, se ha visto que en las zonas rurales no se cuenta con una adecuada tecnificación en la agricultura y mediante el desarrollo del presente proyecto se buscará fomentar el impulso de la investigación y la transferencia tecnológica. Donde se dará a conocer a los productores de cultivos la importancia de la transformación productiva, de manera que puedan alcanzar un crecimiento económico sustentable garantizando la responsabilidad social y ambiental.

Finalmente, este proyecto aporta al cumplimiento de la misión de la Universidad Técnica del Norte que es la de generar, fomentar, y ejecutar procesos de investigación, y de transferencia de saberes, de conocimientos científicos, tecnológicos y de innovación. De igual manera, contribuye a la línea de investigación de "desarrollo de sistemas inteligentes" que está alineada al programa de postgrado en la maestría de telecomunicaciones (Universidad Técnica del Norte, 2018).

Objetivos de Investigación

Objetivo general

Desarrollar un sistema de reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips (*Kakothrips Robustus Uzel*) en los cultivos del guisante o arveja mediante la aplicación de técnicas de DL (*Deep Learning)* con la finalidad de reducir la pérdida del cultivo.

Objetivos específicos

- Revisar bases teóricas sobre las técnicas existentes de DL, utilizadas para la detección y la clasificación de objetos a través del procesamiento de imágenes.
- Construir un repositorio digital del cultivo (guisante o arveja), mediante de la adquisición de imágenes en las parcelas agrícolas.
- Desarrollar un sistema de reconocimiento de lesiones necróticas causadas por la plaga thrips (*Kakothrips Robustus Uzel)* en los cultivos del guisante o arveja, a través del procesamiento y análisis de imágenes.
- Implementar una técnica basada en DL para la detección de la plaga thrips (*Kakothrips Robustus Uzel*) y la clasificación de estado del cultivo, a partir de la extracción y el etiquetado de las características significativas en el análisis de imágenes.
- Validar el sistema propuesto mediante el análisis de los resultados alcanzados tras las pruebas de funcionamiento efectuadas y el criterio de personas involucradas en la producción del guisante o arveja.

Pregunta de Investigación

¿Cómo incide el reconocimiento de lesiones necróticas mediante DL, en la detección de la plaga thrips (Kakothrips Robustus Uzel), en los cultivos del guisante o arveja?

Hipótesis

El desarrollo de un sistema de reconocimiento de lesiones necróticas basado en DL, permite la detección temprana de la plaga thrips (Kakothrips Robustus Uzel) en los cultivos del guisante o arveja.

Preguntas Directrices

¿Qué factores ayudarán a determinar la presencia de la plaga thrips en los cultivos del guisante o arveja?

¿Cuál es la técnica de DL más adecuada para la detección de la plaga thrips en las imágenes del cultivo?

¿Cómo se podría validar la propuesta planteada para la solución del problema presentado?

Variables e indicadores

Independiente: Sistema de reconocimiento de lesiones necróticas.

Dependiente: Detección de la plaga thrips.

Tabla 2

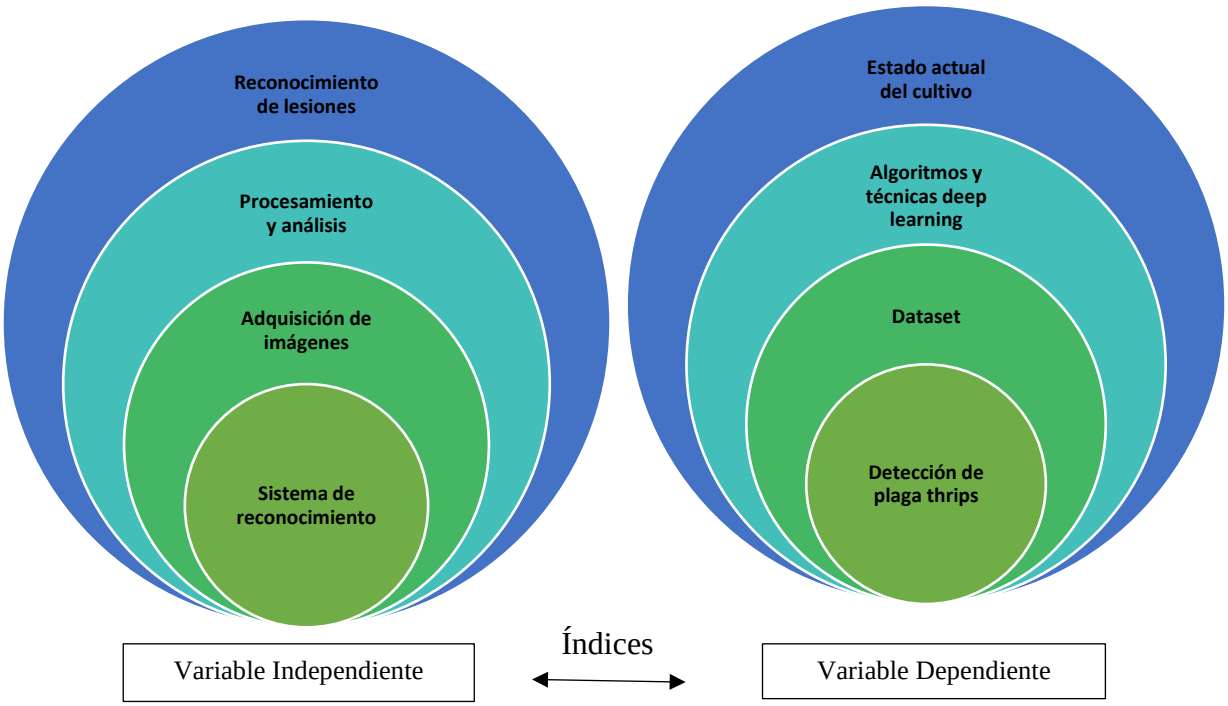
Operacionalización de la Variable Independiente

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS BÁSICOS	TÉCNICA O INSTRUMENTOS
Es un conjunto de métodos y técnicas que mediante algoritmos de visión artificial y técnicas de DL permitirán realizar el análisis y tratamiento de imágenes del guisante	Imágenes	Modelos básicos	¿Qué tipo de imágenes se empleará?	Revisión de documentación bibliográfica
	Segmentación	División en partes fundamentales	¿Cómo se realizará la segmentación para su análisis?	Aplicación de herramientas de Software Libre
	Extracción y Etiquetado	Identificación de características significativas	¿Bajo qué criterios serán identificas las características?	Aplicación de herramientas de Software Libre
	Reconocimiento de lesiones necróticas	Detección de lesiones necróticas	¿Cómo se realizará la detección de lesiones necróticas?	Aplicación herramientas de Software Libre

Nota: Se indican las dimensiones e indicadores de la variable independiente. Realizado por el Autor

Figura 2

Diagrama de correlación de variables



Nota: Diagrama de la correlación entre las variables independiente y dependiente. Realizado por el Autor

Tabla 3

Operacionalización de la Variable Dependiente

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS BÁSICOS	TÉCNICA O INSTRUMENTOS
Plaga que afecta directamente a cultivos o plantas de importancia agronómica.	Plaga thrips	Lesiones necróticas en las hojas, tallo y fruto	¿Qué características presenta la plaga? Nivel del avance de la plaga Estado de los cultivos	Revisión de documentación bibliográfica Visita a los productores para alcanzar un mejor criterio sobre la plaga

Nota: Se indican las dimensiones e indicadores de la variable independiente. Realizado por el Autor

CAPÍTULO II: MARCO REFERENCIAL

Marco Teórico

Antecedentes Investigativos

En Ecuador, específicamente en la provincia del Carchi, la producción del cultivo del guisante o arveja ha presentado un crecimiento indeterminado en zonas comprendidas entre los 2700 y 2800 m.s.n.m., y se ha constituido en uno de los cultivos más importante del sistema de producción del sector (Delgado Chamorro, 2014).

Para la producción y el desarrollo del guisante o arveja se requiere de un clima templado y templado-frío. Este cultivo se puede adaptar a temperaturas bajas en la etapa de germinación e inicial, mientras que para la etapa de desarrollo puede soportar temperaturas entre 16 a 20°C, para su óptimo crecimiento. Pero también, puede soportar temperaturas mínimas de entre 6 a 10°C, y máximas en más de 35°C. Además, requiere de una precipitación media de 500 a 1.000 mm durante todo el periodo vegetativo (Angulo Pérez, 2019).

El cantón Bolívar de la provincia del Carchi, de acuerdo a las condiciones climáticas es una de las zonas consideradas para la producción del guisante o arveja, cuenta con una superficie aproximada de 36.056 hectáreas, con una altura de 2.503 m.s.n.m., y una temperatura promedio entre 12 a 18°C. Los productos transitorios cubren un área de 6.543,47 hectáreas, siendo el fréjol el producto más importante del cantón en la zona del valle, mientras que la arveja, cebolla, cebada, maíz y trigo son los productos

emblemáticos del sector. Siendo la producción de este tipo de cultivos, la principal actividad que más ingresos económicos aporta al cantón. Pero muchos de estos se ven seriamente afectados debido a las condiciones atmosféricas y climáticas que se presentan en la región (GADM-Bolívar, 2015).

En los últimos años se han realizado varios trabajos de investigación sobre la aplicación del aprendizaje profundo o DL en la agricultura, los mismos que muestran algunos aspectos importantes, que pueden aportar y a la vez apalancar el desarrollo de este proyecto de investigación.

Estas investigaciones están relacionadas con sistemas de detección de plagas mediante técnicas de visión artificial; reconocimiento de daños en los cultivos a través de teledetección y aprendizaje automático; clasificación de enfermedades en las plantas a través del uso de redes neuronales convolucionales. Las propuestas anteriores de investigación, permitirán fortalecer el conocimiento científico y técnico de la temática planteada. Con la finalidad de realizar de manera íntegra el desarrollo de un sistema de reconocimiento de lesiones necróticas, para la detección temprana de la plaga thrips en los cultivos del guisante o arveja, mediante técnicas de DL o aprendizaje profundo.

La detección o el reconocimiento temprano de enfermedades o plagas en los cultivos, es una temática que se ha venido estudiando ampliamente a lo largo del tiempo, la misma que continúa siendo de gran interés para varios investigadores.

Fundamentación Legal

El desarrollo de la presente investigación, tiene como sustento legal algunos artículos o lineamientos que rigen a la nación, los que fueron extraídos de las leyes, reglamentos y políticas aprobadas por el Gobierno Nacional, la Asamblea Nacional y Ministerios de la República del Ecuador.

- Artículos de la Constitución de la República del Ecuador (Art. 281, Art. 334, Art. 385, Art. 410).
- Políticas Agropecuarias Ecuatorianas, el MAGAP (Ministerio de Agricultura, Ganadería, Acuicultura y Pesca) establece un conjunto de lineamientos en algunos ejes como: Productividad e Innovación Productiva.
- Artículos de Ley Orgánica de Telecomunicaciones (Art.88).

Esquema del Marco Teórico de la Investigación

El presente trabajo de investigación cubrirá las siguientes temáticas como:

- Detalle rápido sobre la plaga thrips (definición y métodos de control).
- El DL o aprendizaje profundo (definición, técnicas o métodos de DL).
- Arquitecturas de DL para la detección y clasificación de objetos.

Revisión de Investigaciones Afines

En la Tabla 4, se mencionan los artículos científicos afines a la investigación.

Tabla 4

Análisis de artículos afines a la investigación

N°	TEMA	PROBLEMÁTICA/MEJORA	SOLUCIÓN	APORTE A LA INVESTIGACIÖN
1	Detección y clasificación de enfermedades de las plantas mediante aprendizaje profundo (Li, Zhang, & Wang, 2021).	La aplicación del aprendizaje profundo para el reconocimiento de enfermedades en las plantas. Puede evitar las desventajas causadas por la selección artificial. Donde la extracción de características sobre las enfermedades de las plantas sea más objetiva, y mejore la eficiencia de la investigación, así como la velocidad de transformación.	En este artículo, se presentan las tendencias y los desafíos actuales para la detección de enfermedades en las hojas de las plantas mediante DL y técnicas de imágenes avanzadas.	La presente investigación, contribuirá con el desarrollo de un sistema rápido y eficiente. Gracias a que, podrá ser fácilmente parametrizable, para la ejecución de una tarea o acción específica, respecto a otras técnicas y métodos de DL.
2	Clasificación del daño de la fruta de jackfruit usando la red neuronal convolucional (V. Oraño, Maravillas, & G. Aliac, 2020).	Este estudio permite a los agricultores comprender la distribución de plagas y tomar las precauciones adecuadas en tiempo real. El drone agrícola rocía pesticidas solo donde es necesario, lo que reduce el uso de pesticidas, disminuye el daño al medio ambiente y aumenta el rendimiento de los cultivos.	Propone un modelo donde se utilizó una red neuronal convolucional y la implementación de una aplicación móvil en Android. Para la detección y el diagnóstico de los daños de la fruta de jackfruit causados por plagas (barrenador y mosca) y enfermedades (rhizopus y esclerocio).	La investigación aportará con el desarrollo de un sistema de detección de la plaga thrips, el que estará basado en una arquitectura pre-entrenada yolov4-tiny. La que se constituye de múltiples capas convolucionales, que la hace más rápida y efectiva, que sus predecesoras u otros modelos en la detección de objetos.
3	Detección de daños de Brown Planthopper mediante teledetección y aprendizaje automático (Dimuthu, y otros, 2020).	Los productores de arroz, cada año pierden una cantidad significativa en el rendimiento del cultivo, debido a las enfermedades y plagas. El saltamontes pardo (BPH) es una de las enfermedades más comunes que afectan al arroz. El gobierno de Sri Lanka está luchando por hacer estimaciones adecuadas con respecto a la prevalencia del saltamontes marrón debido a la ausencia de datos precisos y oportunos.	Propone un enfoque de aprendizaje automático basado en detección remota de información de un radar de apertura óptica y sintética. La investigación está compuesta de dos fases. En la primera se implementó una clasificación de series de tiempo basada en imágenes SAR, donde se utilizó una red neuronal convolucional para identificar las regiones del arroz cultivadas. En la segunda se utilizó los índices de relación y diferencia estándar, derivados de las imágenes del satélite óptico, a través de una	El sistema permitirá realizar la detección de la plaga thrips en imágenes de diferentes formatos, sin la necesidad de utilizar otras aplicaciones para el análisis y el procesamiento de imágenes. Gracias a que aportará con un dataset robusto y eficiente para la detección específica del objeto de estudio, además podrá ser utilizado en otras investigaciones similares.

			máquina de vectores de soporte para detectar las áreas afectadas por el ataque de la plaga BPH en los cultivos de arroz.	
4	Un sistema de detección de plagas basado en visión inteligente que utiliza un mecanismo de aprendizaje profundo basado en RCNN (Dalai & Senapati, 2020).	La necesidad de controlar plagas y enfermedades, por medios biológicos utilizando tecnología informática e Internet, en lugar de pesticidas para proteger los cultivos es un objetivo primordial de este trabajo.	Para este propósito, se ha implementado y probado un mecanismo de detección basado en RCNN que utiliza segmentación basada en *DL*. El experimento ha demostrado que el enfoque basado en RCNN muestra una mejora significativa sobre el mecanismo común de detección de plagas.	El sistema contará con Yolov4, que es un algoritmo de última generación extremadamente rápido y preciso para la detección de objetos. Este contribuirá a la investigación debido a que permitirá realizar la detección de la plaga en diferentes etapas del cultivo. Siendo un gran aporte para el tratamiento de plagas, gracias a que permitirá reducir el uso de plaguicidas y pesticidas, mediante monitoreos frecuentes a los cultivos.
5	Identificación de plagas de árboles frutales con aprendizaje profundo en drones integrados para lograr una fumigación precisa de pesticidas (Chen, Huang, Li, Chen, & Cha, 2021).	Este estudio permite a los agricultores comprender la distribución de plagas y tomar las precauciones adecuadas en tiempo real. El drone agrícola rocía pesticidas solo donde es necesario, lo que reduce el uso de pesticidas, disminuye el daño al medio ambiente y aumenta el rendimiento de los cultivos.	En este estudio se aplican nuevas aplicaciones de inteligencia de bordes para establecer un sistema inteligente de reconocimiento de plagas para manejar este problema. Se usa un drone de detección para fotografiar la plaga y empleamos un modelo de red neuronal Tiny-YOLOv3 construido en un sistema integrado NVIDIA Jetson TX2 para reconocer papillosa en el huerto y determinar la posición de las plagas en tiempo real.	La propuesta del sistema, se basará en una red neuronal convolucional, la que permitirá efectuar una verdadera interpretación del objeto de estudio. Este sistema contribuirá a la investigación, debido a que utilizará yolov4-tiny, para la ejecución de las actividades sobre dispositivos que tengan limitados recursos tecnológicos como los SBC (Single Board Computer).

6	Investigación sobre el modelo de reconocimiento de enfermedades de los cultivos y plagas de insectos basado en el aprendizaje profundo en entornos hostiles (Ai, Sun, Tie, & Cai, 2020).	Las enfermedades agrícolas y las plagas de insectos son uno de los factores más importantes que amenazan seriamente la producción agrícola. La detección e identificación tempranas de plagas puede reducir eficazmente las pérdidas económicas causadas por las plagas.	Se propone un modelo de red neuronal convolución, que se utiliza para identificar automáticamente las enfermedades de los cultivos. Los datos provienen de un conjunto de datos públicos de la competencia AI Challenger en 2018, con 27 imágenes de enfermedades de 10 cultivos. En este documento, se utiliza el modelo Inception-ResNet-v2	El diseño del sistema de detección aportará a las investigaciones relacionadas, ya que asegurará el funcionamiento de la red yolov4-tiny, en condiciones adecuadas e incluso cuando las condiciones de los entornos experimentales presentan cambios bruscos.
7	Modelos de predicción para la identificación y el diagnóstico de enfermedades de las plantas de tomate (Verma, Chug, & Singh, 2018).	A lo largo de los siglos, las plagas y enfermedades de las plantas han seguido siendo una amenaza constante para la calidad y cantidad de la producción agrícola en general. Por lo tanto, su identificación oportuna y precisa podría atenuar en gran medida las pérdidas económicas en todo el mundo, denigrando también el impacto nocivo de los fertilizantes y pesticidas en el medio ambiente.	Se presenta una encuesta sobre las técnicas y modelos de predicción actuales, basado en el procesamiento de imágenes y el papel de Internet de las cosas (IoT), que se aplica para la identificación, detección y cuantificación de las enfermedades de las plantas del tomate.	El presente trabajo contribuirá a la investigación, debido a que propone un sistema de detección de objetos que puede ser adaptado con las nuevas tendencias tecnológicas, gracias a que utilizará un algoritmo de última generación rápido y eficiente.

Nota: En la tabla se muestran los diferentes artículos científicos afines al tema de investigación que fueron analizados y servirán como referencia de la presente investigación.

Realizado por el Autor.

Plaga Thrips

Los thrips constituyen un grupo diverso de especies, estos son pequeños insectos su contextura es de forma cilíndrica, alargada y su color puede variar del gris al amarillo o al marrón, pertenecen a la familia de los tisanópteros[2] y muchos de ellos de importancia agrícola, pueden desplazarse por el cultivo realizando pequeños vuelos o llevados por el viento. Por su hábito alimenticio estos insectos son chupadores que suelen estar en los tallos, hojas, flores y en el polen de las plantas en una variedad de cultivos, en donde se alimentan y se reproducen (Toledo Perdomo & Sagastume Mena, 2018).

Figura 3

Thrips del guisante o arveja (Kakothrips Robustus Uzel)

Nota: Plaga thrips que ataca al guisante o arveja. Tomado de TRIPS DEL GUISANTE - KAKOTHRIPS ROBUSTUS [Imagen], Agronomija, www.agronomija.rs

Daños de la Plaga Thrips

Los daños provocados por thrips, en el cultivo del guisante o arveja, pueden ser detectados principalmente en las vainas por las cicatrices que éstas tienen. También,

[2] Tisanópteros: son pequeños insectos, llamados a veces thrips, thrips o arañuelas que se presentan en los cultivos.

existe la presencia de síntomas como la aparición de manchas de color plateado en los tallos, hojas, flores. Debido a que, al extraer los fluidos en las células vegetales, estas se llenan de aire y su aspecto se vuelve de color gris plateado y se observa puntos negros en las áreas afectadas que es el excremento de plaga. En muchas de las ocasiones estos síntomas provocan la caída de las hojas del cultivo, en cambio que la cicatriz en las vainas causa el rechazo de la calidad del producto (Prado Jiménez, 2019).

Métodos de Control de la Plaga Thrips

Para control de la plaga thrips en los cultivos se utilizan los siguientes métodos de control.

Trampeo: según (Prado Jiménez, 2019), en su investigación señala las siguientes indicaciones para reducir la presencia de la plaga thrips en el guisante o arveja:

- Instalación de mallas en las bandas del invernadero y vigilar que no existan roturas en el plástico.
- Desherbar dentro y fuera del invernadero y eliminación de restos de cultivo, antes de realizar una nueva plantación y distanciamiento de tiempo posible de la anterior.
- Implementación de trampas adhesivas antithrips, desde el inicio del cultivo hasta el final del mismo, para realizar un seguimiento al guisante o arveja.

Control etológico: es la instalación de trampas amarillas, de acuerdo a las evaluaciones realizadas por el proyecto ICTA-CATIE-ARF. Determinaron que las plagas

eran atraídas hacia este tipo de trampas, deben ser untadas con algún agente pegajoso y la mezcla de algunos componentes como vaselinas sólida y líquida en una proporción 1:1, además recomiendan el uso de aceite de motor y varios productos industriales (Prado Jiménez, 2019).

Control biológico: Según infoAgro (s.f.) en una investigación sobre el control biológico de la plaga thrips menciona que:

Los productos biológicos para el control de frankliniella-occidentalis[3] o thrips, son los compuestos a base del hongo verticillium-lecanii[4] y productos de sales potásicas de ácidos grasos. El hongo verticillium-lecanii no es nocivo para los enemigos naturales, de modo que puede ser utilizado para suplementar el control cuando los ácaros y los chinches depredadores no logran controlar la plaga completamente. (infoAgro, s.f.)

Control químico: en el control químico, se lo realiza a toda la planta, especialmente en las hojas, flores y frutos. Por lo general realiza dos tratamientos químicos cada 7 días, con compuestos químicos como el metiocarb, fenitrotión, y malatión en dosis de 200 g/200 litros de agua (infoAgro, s.f.).

Control por sistemas automáticos: en la actualidad se han desarrollado algunos sistemas automáticos para el control de la plaga a través del uso de diferentes tecnologías como son los de dispersión de plaguicidas sobre áreas donde se encuentra la plaga.

[3] Frankliniella-Occidentalis: nombre común de los thrips de las flores que presentan en los cultivos como el vid, cítricos y especies ornamentales.
[4] Verticllium-Lecanii: es un hongo entomatógeno muy usado en el mundo para el control biológico de mosca blanca y pulgón (en menor medida thrips y araña roja).

En la actualidad existen varias tecnológicas que están impulsando el despliegue de la agricultura de precisión a través del desarrollo de aplicaciones basadas en IoT (*Internet of Things*) para la gestión de la información de los sensores, protocolos de MQTT (*Mesaré Queue Telemetry Transport*) para la mensajería de los datos y métodos de IA (*Artificial Intelligence)* asentados en aprendizaje automático para la detección de frutos a partir del procesamiento de imágenes.

Deep Learning

El aprendizaje profundo o también conocido como DL, es un campo del aprendizaje automático o ML (Machine Learning), que a la vez es parte esencial de la IA. La que tiene como objetivo fundamental ofrecer técnicas y algoritmos que permitan resolver ciertos inconvenientes que los seres humanos pueden solventar de manera automática o intuitiva (Oliva Rodríguez, 2018).

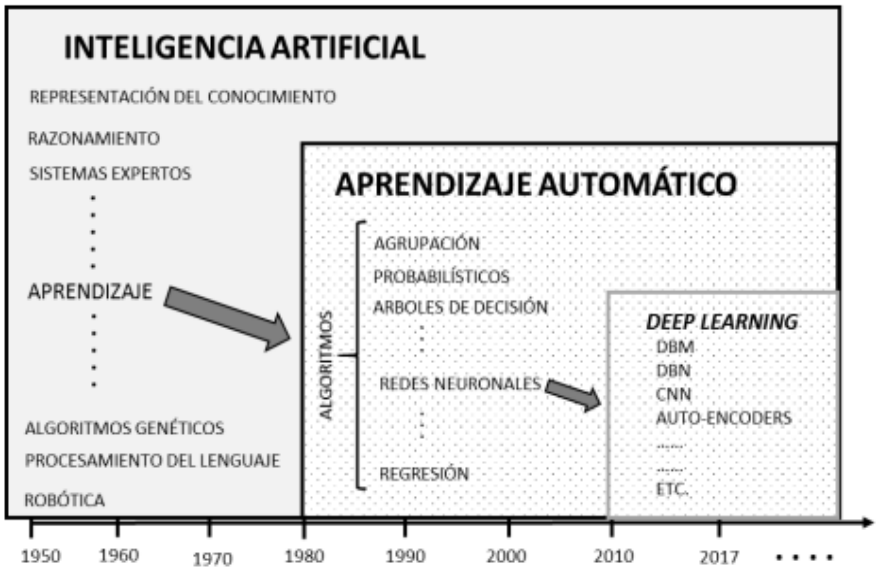
> El DL, es un conjunto de algoritmos que tienen la capacidad de modelar abstracciones de alto nivel mediante el uso de varias capas de procesamiento. Este campo de investigación inició en los años 80 y es una rama del aprendizaje automático. Donde sus algoritmos son empleados para el entrenamiento de las redes neuronales profundas con la finalidad de alcanzar mayor precisión que el ML. (Suntaxi Sarango, 2019, p.12)

La precisión obtenida por el DL es muy eficiente, que en ocasiones ha llegado a superar al ser humano. Por ejemplo, el aprendizaje profundo ha sido utilizado en para

investigaciones médicas, conducción autónoma, automatización industrial entre otras (Sánchez Martínez, 2018).

Figura 4

Evolución del deep learning



Nota: Inteligencia artificial y los diferentes tipos de evolución que ha tenido el aprendizaje. Tomado de *Situación del deep learning dentro de la Inteliegencia Artificial* [Imagen], Sánchez Martínez, 2018, www.oa.upm.es.

Según Suntaxi Sarango, menciona que: "Además del desarrollo de mejores algoritmos, existen dos razones por las que resulta útil emplear modelos de redes profundas" (2019, p.12). Las principales razones de emplear redes profundas son:

- Datos etiquetados: El entrenamiento de modelos DL, necesita de una gran cantidad de información etiquetada. Por ejemplo, para el desarrollo de un vehículo autónomo se requieren millones de imágenes y miles de videos respectivamente etiquetados.

- Potencia de cálculo: Un modelo DL requiere de una potencia de cálculo significativa. Actualmente existen tecnologías como las GPU[5] (Unidad de Procesamiento Gráfico), programación paralela, clústers y cálculo en la nube las que han permitido disminuir el tiempo del entrenamiento de una red neuronal profunda.

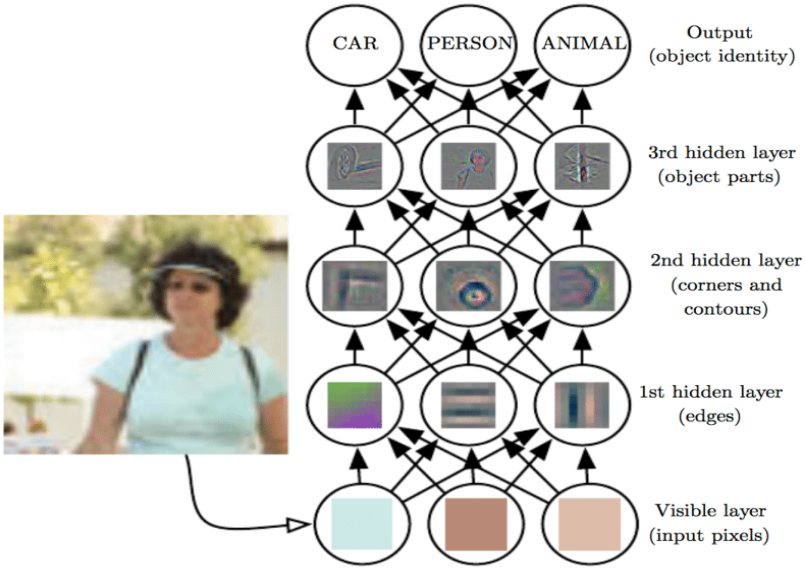
El DL pretende abordar los problemas más intuitivos, donde propone que las máquinas aprendan de su experiencia y lleguen a percibir el mundo en niveles de concepción. Donde, cada concepción se relaciona a otras más simples. Este tipo de aprendizaje tiene la capacidad de aprender la representación de información, combinando los niveles de concepción. De esta manera, aprende las representaciones más complejas basándose en otras simples (González Muñiz, 2018).

Los algoritmos de DL, permiten separar los factores variables de la información observada. Además, tiene la capacidad de extraer las características significativas y abstractas de los datos originales con la finalidad de analizar de una forma eficiente el objeto de estudio como se observa en la Figura 5. Las redes neuronales profundas de manera computacional son arquitecturas complejas que se obtienen de la interconexión de múltiples unidades simples o neuronas artificiales. Dentro del DL, existen diferentes métodos o técnicas como autoencoders, redes convolucionales, redes de creencia profunda y redes recurrentes (González Muñiz, 2018).

[5] GPU: es el acrónimo de Unidad de Procesamiento Gráfico y representa precisamente el corazón de una tarjeta gráfica al igual que la CPU lo hace en un PC.

Figura 5

Modelo de aprendizaje profundo o DL



Nota: Estructura general de un modelo de aprendizaje profundo o DL. Tomado de *Ilustración de un modelo deep learning* [Imagen], González Muñiz , 2018, www.deeplearningbook.org.

Métodos de Deep Learning

Entre los métodos o técnicas más utilizadas para la detección de objetos en tiempo real están las redes neuronales convolucionales.

Redes Neuronales Convolucionales (CNN)

Las CNN son redes neuronales que realizan de manera precisa tareas como la detección y clasificación de objetos, además de la segmentación de imágenes. Estas redes tienen una estructura similar a las redes neuronales ordinarias. Donde la principal diferencia es la aparición de capas especializadas para la extracción de características significativas de una imagen de forma automática (Gama García, 2020).

De acuerdo a lo que expresa Sánchez Martínez, en su investigación:

> Las CNN se caracterizan por realizar operaciones de convolución entre una capa de entrada y un filtro o kernel, obteniendo como resultado un mapa de características. El proceso se repite en las capas intermedias, con el objetivo de extraer características significativas con distintos niveles de abstracción. La capa de salida realiza la clasificación de objeto analizado. De esta manera, la red convolucional actúa como extractor y clasificador de características. Donde cada neurona de una determinada capa no recibe información de todas las neuronas de la capa anterior, sino solo de algunas. Esto hace que las neuronas reduzcan el número de operaciones necesarias. (2018, p.18)

La convolución es una función matemática que se obtiene del resultado la superposición de una función *f* (*t*), y una de función que es la trasladada e invertida de otra función *g* (*t*). En la Ecuación 1 se muestra la función de la operación de convolución (Sánchez Martínez, 2018).

Ecuación 1

Función de la operación de convolución

$$(f * g)(t) = \int_0^t f(t-\tau)g(\tau)d_\tau$$

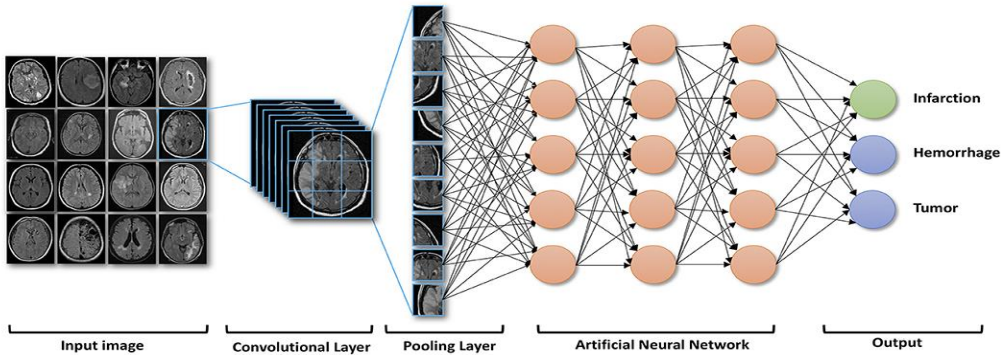
Se menciona que las capas iniciales de las redes CNN, tienen la capacidad de realizar la detección de algunas formas simples como líneas y curvas, las mismas que se

van perfeccionando hasta alcanzar a las capas más profundas, donde se llega a reconocer formas complejas como los rostros de las personas (Gama García, 2020).

La arquitectura de una CNN, se compone básicamente de tres tipos de capas que son entrada de datos, extracción de características y clasificación como se observa en la Figura 6.

Figura 6

Arquitectura de red de neuronal convolucional



Nota: Arquitectura completa de una red neuronal convolucional. Tomada de *Computer Neural Network* [Imagen], frontiers, 2019, www.frontiersin.org.

Capa de entrada de datos: Es la capa que recibe los datos que provienen del exterior de la CNN, la que suele ser representada en una cuadrícula tridimensional en la que sus dimensiones dependen de los parámetros de la Ecuación 2.

Ecuación 2

Valor de la capa de entrada de datos

$$I = W * H * D$$

Donde las dimensiones de la cuadrícula dependen del ancho de la imagen *(W)*, de la altura de la imagen *(H)* y de la profundidad o el número de canales de la imagen *(D)* (Maeda Gutiérrez, 2019).

Capa de extracción de características: Esta es una serie de capas que son las encargadas de obtener la información para construir progresivamente las características de orden superior. Esta se capa se clasifica en capas de convolución que calcula la función y capas de pooling que reúnen los resultados obtenidos de las capas convolución (Maeda Gutiérrez, 2019).

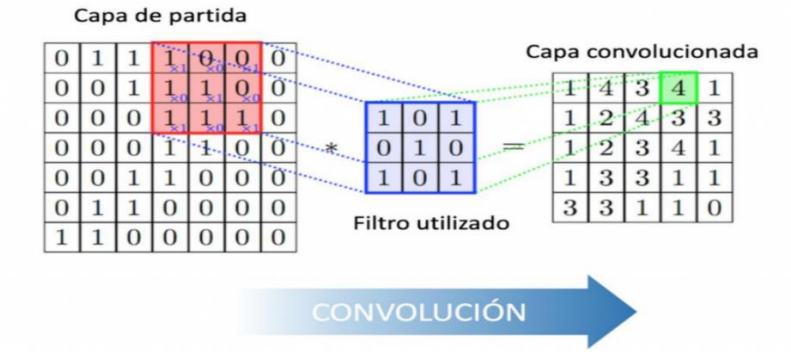
Capa convolución: La capa convolución es la parte más importante de una CNN, esta tiene la capacidad de transformar la información de entrada mediante una serie de neuronas las que se conectan de forma local desde la capa anterior. Esta capa ejecutará el producto entre una región específica de la capa de entrada y los pesos de la capa de salida, este proceso mantiene las mismas dimensiones espaciales. El reconocimiento de las características del objeto en las CNN, se realiza mediante la operación de convolución. Donde la entrada de estas capas pueden ser los mismos valores de inicio y la salida son valores que de otra convolución anterior (Maeda Gutiérrez, 2019).

La convolución es la aplicación de un kernel o filtro a la imagen de entrada, el mismo que se desplaza por toda la imagen de izquierda a derecha y desde arriba hacia abajo. De esta manera, realiza el cálculo del producto entre cada uno de los elementos de la imagen y del kernel en esa posición, de forma que se puedan sumar todos los resultados obtenidos de la convolución (Gama García, 2020).

En la Figura 7 se muestra como el kernel se desplaza por cada uno de los valores de entrada y en cada paso multiplica el kernel por los dichos valores, donde se crea una nueva característica de salida.

Figura 7

Representación de la operación de convolución



Nota: Operación de convolución con un filtro de 3x3. Tomada de *Ejemplo de una convolución* [Imagen], Berrondo Urruzola, 2020, www.addi.ehu.es.

Cada kernel mediante la convolución crea un mapa de características de tamaño (WH+ 1) x (W-H+1) x P.

Donde P, es el número de kernels o filtros que se utilizarán en la operación. En cambio, los resultados de la convolución son almacenados en la matriz de activación. Una vez que, el kernel haya recorrido toda la imagen (Pérez Lorenzo, 2019).

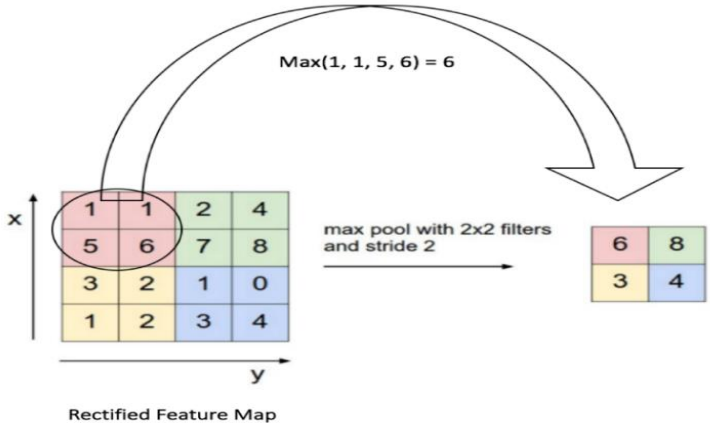
Capas de pooling: estas capas por lo general se encuentran a continuación de la capa de convolución, y son las encargadas de eliminar el volumen de parámetros a analizar mediante la reducción de dimensiones espaciales (alto y ancho) de la cantidad de entradas para la siguiente capa convolucional. Esta reducción implica la pérdida de

información; pero permite tener una menor carga computacional en el cálculo de las próximas capas reduciendo así el sobre ajuste de las mismas (Suntaxi Sarango, 2019).

Este tipo de capas utiliza un filtro de 2x2, de modo que los cuatro píxeles que serán filtrados son sustituidos por un solo valor, que corresponde al máximo valor de los píxeles filtrados como se observa en la Figura 8 (Jiménez Silva, 2018).

Figura 8

Diagrama de aplicación de max-pooling



Nota: Operación de max-pooling para un filtro de 2x2. Tomado de *Max Polling* [Imagen], scrapbook, 2016, www.stephanosterburg.gitbook.io.

Según Gama García (2020), en su investigación menciona que existen tres formas principales de aplicar la operación de pooling

- "Max-pooling: devuelve el máximo valor de entre todos los píxeles cubiertos por el kernel". (p.12)
- "Min-pooling: devuelve el valor mínimo de entre todos los píxeles cubiertos por el kernel". (p.12)

- “Average pooling: se calcula el valor medio de entre todos los píxeles cubiertos por el kernel”. (p.12)

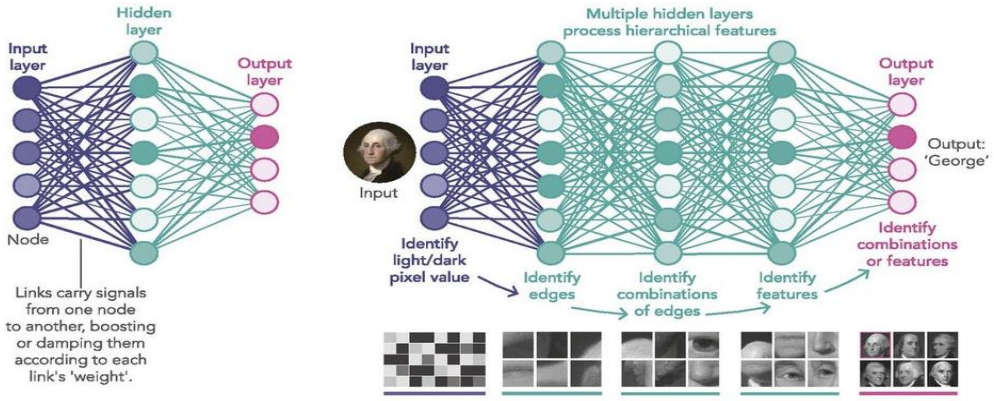
Por ejemplo, si la matriz de entrada a la capa pooling es de tamaño (W x H x D), la matriz de salida será (W/P x H/P x D), al emplear pooling se reducen las dimensiones de la matriz de activación y, por ende, disminuye el número de parámetros y el tiempo computacional (Pérez Lorenzo, 2019).

Capa de clasificación: la capa de clasificación es la última de las CNN, este tipo de capas se encuentran conectadas a sus respectivas capas adyacentes. Donde la función principal es tomar las características finales que fueron calculadas por la red y además genera probabilidades correspondientes a las clases entrenadas. Los resultados de estas capas representan los datos de salida, los que tendrán menores dimensiones, siendo representados en un vector que contiene ciertos elementos como clases que hayan sido analizadas (Jiménez Silva, 2018).

En la Figura 9 se indica un diagrama de las capas totalmente conectadas para la clasificación de un objeto en la CNN.

Figura 9

Diagrama de capas totalmente conectadas



Nota: El diagrama muestra las capas de inicial, intermedias (ocultas) y la final las mismas que se conectan totalmente. Tomado de *Diferencia entre redes neuronales simples y profundas* [Imagen], FutureSpace, 2021, www.futurespace.es/wp-content/uploads/2021/03/deeprrnn.jpg

Las capas clasificadoras tendrán la misma cantidad de neuronas, que de clases a predecir. Normalmente estas capas utilizan la función Softmax[6], que devuelve valores de salida, de manera que la suma de estos valores sea igual a 1 (Sánchez Martínez, 2018).

Arquitecturas para la Clasificación de Imágenes

Las CNN son la mejor opción para la clasificación de objetos en tiempo real debido a que es una de las arquitecturas más eficientes para este tipo de tarea. Pero la combinación de capas a nivel de extensión y profundidad en los modelos más complejos es una de las partes difíciles en el uso de las CNN. A continuación, se detallan las principales arquitecturas utilizadas para la clasificación de objetos.

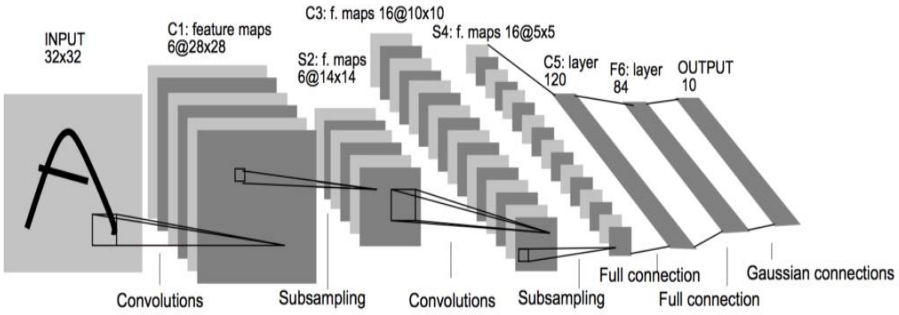
[6] Softmax: es una función matemática que convierte un vector de números en un vector de probabilidades, donde las probabilidades de cada valor son proporcionales a la escala relativa de cada valor en el vector.

Arquitecturas Clásicas

LeNet-5: La arquitectura LeNet como se indica en la Figura 10, fue creada por LeCun en 1998, esta es una red que fue utilizada como la primera CNN. Esta arquitectura fue inicialmente creada para identificar caracteres escritos a mano y en máquina. La red LeNet-5 se compone básicamente de dos capas convolucionales y de algunas capas average pooling (Berrones Reyes, 2019).

Figura 10

Diagrama de la arquitectura LeNet-5



Nota: Arquitectura total de LeNet-5 para la clasificación de objetos. Tomada de Imagen original de LeNet-5 [Imagen], Ichi.Pro, s.f., www. ichi.pro/es/lenet-5-en-kotlin-con-tensorflow-202630331086503

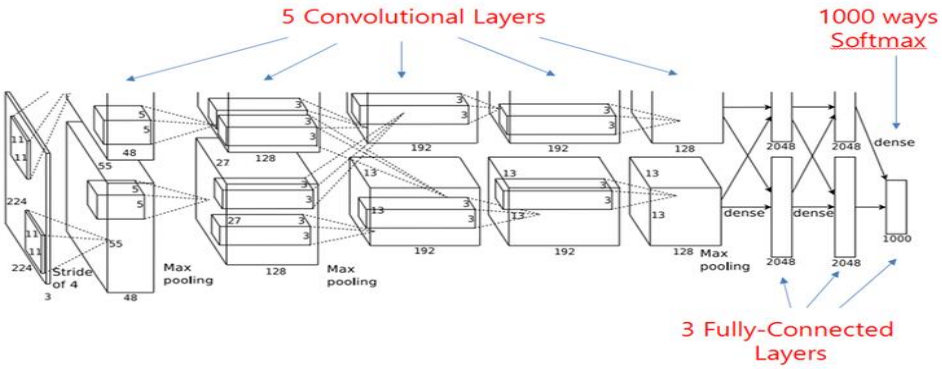
AlexNet: la arquitectura AlexNet como se muestra en la Figura 11, fue creada por Alex Krizhevskey, Ilya Sutskever y Geoffrey E. Hinton en 2012. Esta red CNN, fue entrenada para clasificar 1.3 millones de imágenes de alta resolución sobre un conjunto de entrenamiento del ImageNet divididas en 1000 clases distintas. Esta red CNN, está formada de 60 millones de parámetros y 500000 neuronas distribuidas en cinco capas convolucionales, las que se conectan de manera directa a la capa de maxpooling[7] y a las

[7]Capa Maxpooling: Son múltiples capas de filtros convolucionales de una o más dimensiones.

capas conectadas de manera global, y finalmente a un softmax de 1000 caminos. Para tratar de eludir el sobre ajuste en los datos se utiliza el método de regulación dropout[8], que ha sido muy eficiente; pero incrementa el procesamiento y el tiempo computacional (Maeda Gutiérrez, 2019).

Figura 11

Diagrama de la arquitectura AlexNet



Nota: En la figura se muestra la arquitectura total de AlexNet. Tomada de *Clasificador AlexNet* [Imagen], Manoharan , 2017, www.github.com/narenkmanoharan/ImageNet-Classifier-Tensorflow

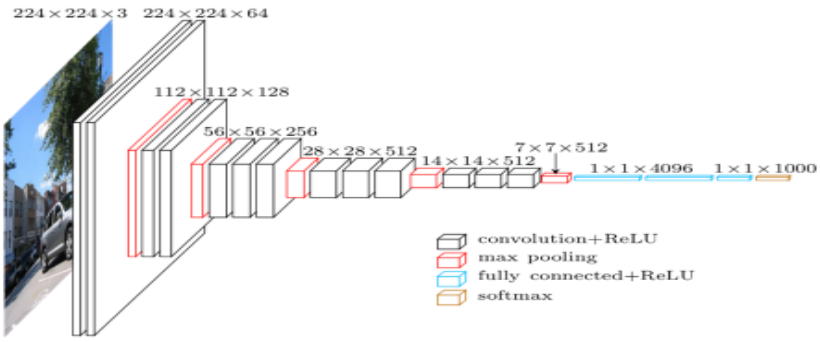
VGGNet: la arquitectura VGGNet como se indica en la Figura 12, fue creada por Simonyan y Zisserman en el 2014. Esta red tiene algunas versiones entre las que se destacan las versiones VGG-16 y VGG-19 para el reconocimiento de imágenes. La red VGGNet se caracteriza por ser una arquitectura compacta de capas convolucionales 3x3 puestas una sobre otra, consiguiendo así un mayor nivel de profundidad, además en dos de las capas totalmente conectadas se encuentra un clasificador softmax. Esta arquitectura es una de las más utilizadas, debido a la capacidad que tiene para la extracción de

[8]Regulación Dropout: Es una técnica de regularización que es aplicada en base a la probabilidad dada por la distribución de Bernoulli

características. Sin embargo, esta red tiene más de 140 millones de parámetros, que la hace compleja su manipulación (Berrones Reyes, 2019).

Figura 12

Diagrama de la arquitectura VGGNet



Nota: Arquitectura de VGGNet para la clasificación de objetos. Tomada de *Arquitectura VGGNet* [Imagen], ResearchGate, 2019, www.researchgate.net/figure/VGGNet-architecture-19_fig2_333242381

Arquitecturas Modernas

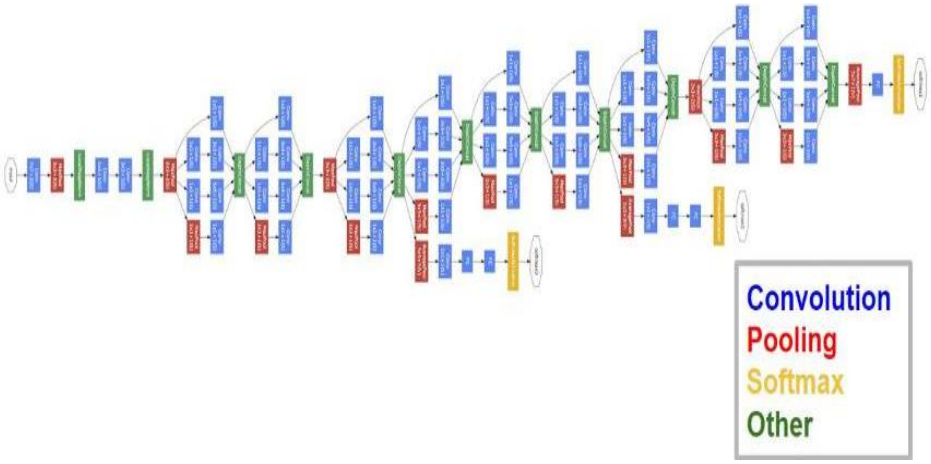
GoogLeNet: la arquitectura GoogLeNet, como se indica en la Figura 13, fue creada por Szegedy en el 2015. Esta fue inspirada en la LeNet-5, gracias a que se añadieron algunos elementos que ayudaron a mejorar la jerarquía de profundidad de la red. GoogLeNet utiliza la normalización de parches y el optimizador de RMSProp[9].Además, disminuyó el número de parámetros, gracias a que utiliza convoluciones pequeñas. En comparación a la red AlexNet que utiliza 60 millones de parámetros, GoogLeNet emplea solamente 4 millones de parámetros (Berrones Reyes, 2019).

[9] RMSProp: es una técnica de optimización basada en gradientes que se utiliza en el entrenamiento de redes neuronales.

GoogLeNet contiene nueve módulos iniciales, cuatro capas convolucionales, cuatro capas de agrupación máxima, tres capas de agrupamiento promedio, cinco capas completamente conectadas y tres capas de softmax para los principales clasificadores auxiliares de la red. Además, utiliza la regularización de dropout en la capa totalmente conectada y aplica la activación ReLU[10] en todas las capas convolucionales. (Maeda Gutiérrez, 2019, p.18)

Figura 13

Diagrama de la arquitectura GoogLeNet



*Nota***:** Arquitectura completa de GoogLeNet para la clasificación de objetos. Tomada de *GoogLeNet / Inception* [Imagen], Siddharth Das, 2017, www.medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

ResNet: la arquitectura ResNet fue creada por He en el año 2015, esta estructura como se muestra en la Figura 14, se basa en el paso de conexiones y características más significativas, con el objetivo de normalizar los parches del objeto de estudio. “El paso de conexión se puede describir como unidades cerradas o unidades recurrentes cerradas

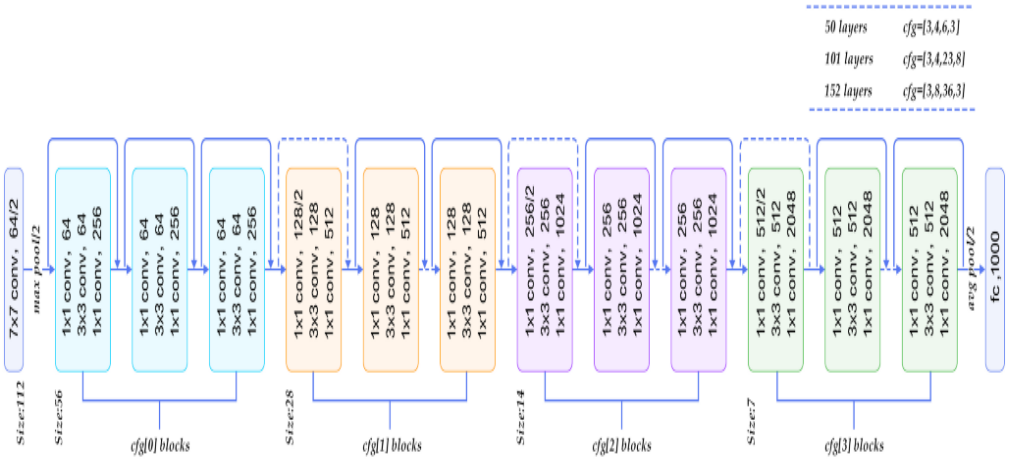
[10] ReLU: es la función de activación del rectificador o ReLU (Unidad lineal rectificada, esta función comenzó a aparecer en el contexto de la extracción de características visuales en redes neuronales jerárquicas a partir de finales de la década de 1960.

que han sido aplicadas recientemente en las redes neuronales recurrentes" (Berrones Reyes, 2019, p.39).

Por ejemplo, esta arquitectura fue entrenada con 152 capas donde se obtuvo menor complejidad computacional en comparación a la red VGG, con un porcentaje de error del 3.57%. De esta manera, se pudo comprobar que el rendimiento es superior al de una persona respecto a la identificación y procesamiento de imágenes (Berrones Reyes, 2019).

Figura 14

Diagrama de la arquitectura ResNet



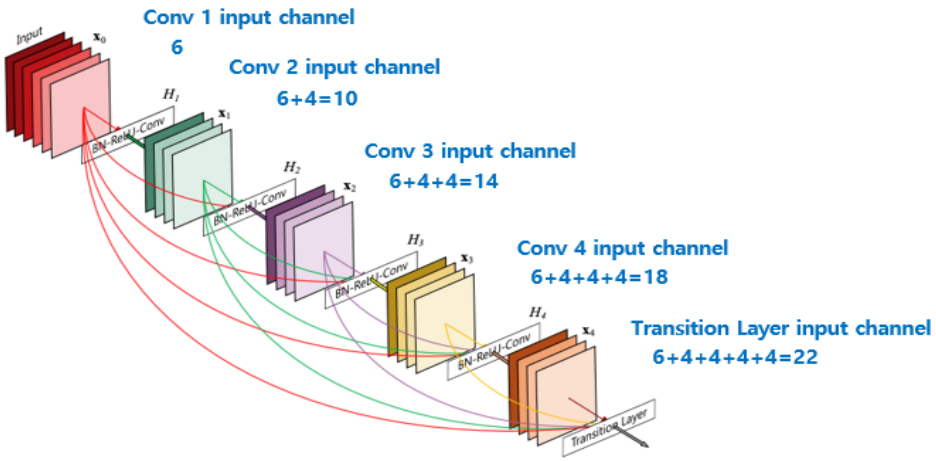
***Nota*:** Arquitectura de completa de ResNet para la clasificación de objetos. Tomada de *Diagrama de arquitectura ResNet* [Imagen], Utrera Burgal, 2018, www.jesusutrera.com/articles/article07.html

DenseNet: la arquitectura DenseNet fue creada por Huang en el 2017, es una extensión del modelo de la red ResNet. Donde se añaden conexiones residuales no solo a la salida del módulo sino a todos los consecutivos como se indica en la Figura 15.

Donde, el mapa de características de cada capa se concatena a la entrada de cada capa sucesiva dentro de un bloque denso. Esta acción permite que las capas posteriores de la red usen directamente las características de las capas anteriores, impulsando así la reutilización de las características dentro de la red. De esta manera, se ratifica que la concatenación de los mapas de características aprendidas por las diferentes capas incrementa la variación en la entrada de las capas posteriores, mejorando así la eficiencia. (Ordóñez Ramos, 2020, p.21)

Figura 15

Diagrama de la arquitectura DenseNet



Nota: Arquitectura completa de DenseNet para la clasificación de objetos. Tomada de *A5- layers dense block with a growth rate of k=4* [Imagen], yw0nam, 2020, www.github.com/yw0nam/DenseNet

Arquitecturas para la Detección de Imágenes

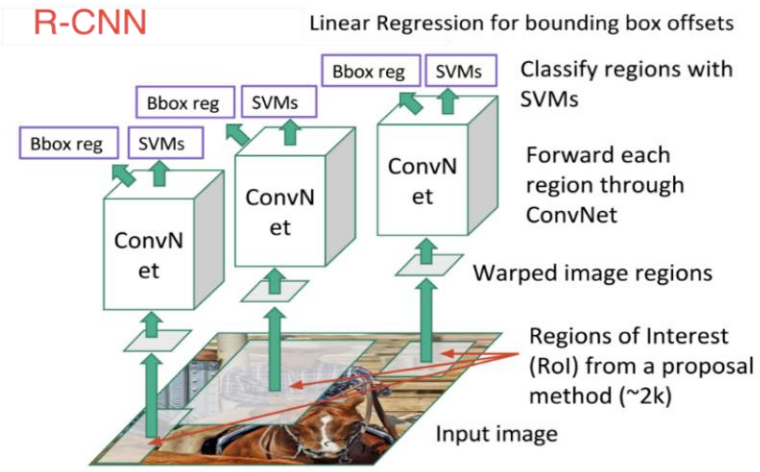
Dentro de las redes CNN, se han desarrollado algoritmos que se basan en R-CNN, u otros como Yolo, los que permiten acelerar el procesamiento, tanto para la localización

como para el reconocimiento de objetos a continuación se realizará una explicación de cada uno de estos algoritmos.

R-CNN: las redes CNN basadas en la región (R-CNN), como se muestra en la Figura 16 se basan en realizar una búsqueda selectiva que consiste en extraer un número fijo de propuestas de la región en la imagen original. Esta arquitectura se conforma en cuatro etapas, la primera es la imagen de entrada, la segunda es el método de búsqueda selectiva. Donde se clasifican las imágenes en un conjunto de hasta 2000 regiones, siendo cada una de estas un objeto, para este tipo de propuestas se calcula un vector de 4096 dimensiones, el que representa la salida de información. En la tercera etapa se emplea un modelo de red CNN entrenado, que efectúa la extracción de características en la imagen. En la última etapa se utilizará un SVM (Support Vector Machine), el que clasificará la presencia del objeto en la región seleccionada (Burgués Miró, 2019).

Figura 16

Diagrama de la arquitectura R-CNN

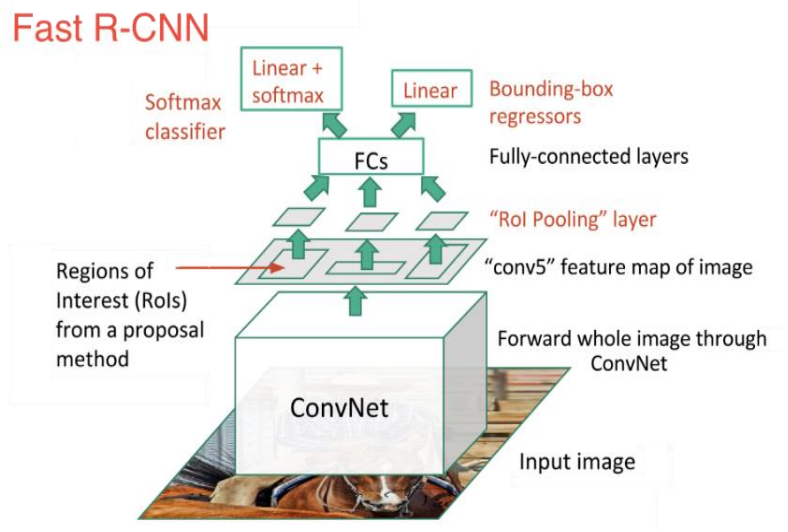


*Nota***:** Arquitectura de R-CNN para la detección de objetos. Tomada de *R-CNN* [Imagen], Xu, 2018, www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos

Fast R-CNN: la arquitectura Fast R-CNN, tiene el mismo funcionamiento que R-CNN, con la diferencia de solventar los problemas que se presentan en la R-CNN. Donde no solo se involucra la rapidez para la detección de objetos, además que la precisión media es más alta. La principal diferencia respecto a la R-CNN es que toda la imagen de entrada se alimenta a la CNN, en la que se crea un mapa de características convolucionales. A continuación, las propuestas son llevadas del mapa y transformadas en cuadros mediante una capa de agrupación de la región de interés (RoI) como se indica en la Figura 17 (Burgués Miró, 2019).

Figura 17

Diagrama de la arquitectura Fast R-CNN



Nota: Arquitectura de Fast R-CNN para detección de objetos. Tomada de *Fast R-CNN* [Imagen], Xu, 2018, www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos

La capa RoI,[11] efectúa una agrupación máxima para reducir la dimensión de las entradas, que son los mapas de las características de la CNN, en la que genera un pequeño

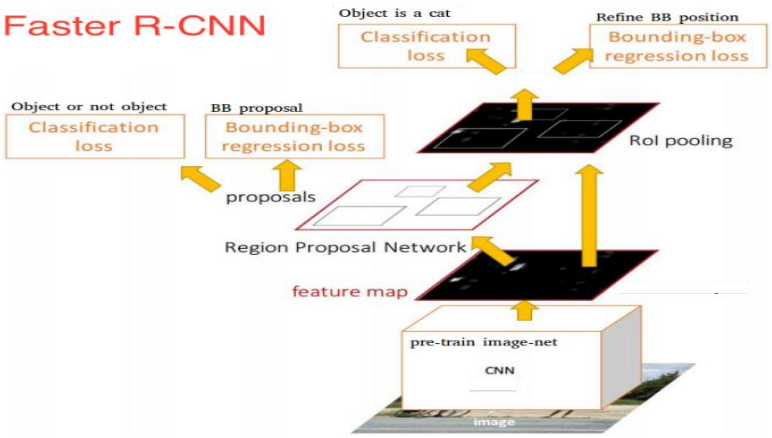
[11] RoI: Región de la agrupación de interés (también conocido como ROI puesta en común) es una operación ampliamente utilizada en tareas de detección de objetos mediante redes neuronales convolucionales.

mapa de tamaño fijo, para que ingresen a una capa totalmente conectada (FC). A partir del vector de características generado por la capa RoI, se utiliza una capa Softmax, que ayuda a decidir a qué clase pertenece el objeto, mientras que con la capa Bounding Box se obtienen las coordenadas de desplazamiento para el marco delimitador de las clases (Torres Alonzo, 2020).

Faster R-CNN: con la mejora significativa en tiempo de respuesta que tiene el algoritmo Fast R-CNN respecto al R-CNN, los dos utilizan la búsqueda selectiva para encontrar las propuestas de región. El algoritmo utilizado por los anteriores modelos es lento y requiere de mucho tiempo para su ejecución, además afecta el comportamiento de la red. Debido a las circunstancias fue creado un algoritmo para la detección de objetos, eliminando el algoritmo de búsqueda selectiva dejando que la red aprenda de las propuestas de región como se muestra en la Figura 18 (Burgués Miró, 2019).

Figura 18

Diagrama de la arquitectura Faster R-CNN



*Nota***:** Arquitectura de Faster R-CNN para la detección de objetos, Tomado de *Faster R-CNN,* [Imagen], Xu, 2018, www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos

De igual manera, el modelo Fast R-CNN la imagen de entrada se alimenta a la CNN, y crea un mapa de características convolucionales. Donde, la principal diferencia es la utilización de una red de propuesta de región (RPN), que es empleada para predecir las propuestas de la región. Las regiones pronosticadas de una imagen son redimensionadas, mediante la capa de agrupación RoI, que es usada para clasificar la imagen entre la región propuesta. Pronosticando así los valores de los cuadros delimitadores, para este tipo de estructuras el periodo de detección es de aproximadamente 0,2 segundos (Burgués Miró, 2019).

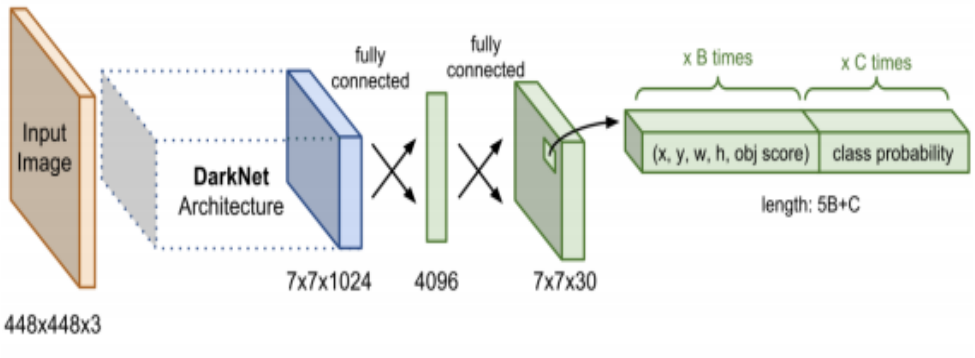
Esta arquitectura para la detección de objetos se basa en propuestas de regiones con una alta tasa de aciertos y tiempos de ejecución bastante mínimos, por lo que cuanto más actual es la versión del algoritmo este es más rápido y preciso. Pero tiene un gran inconveniente al momento de utilizarlo para el procesamiento de imágenes en la detección de objetos en tiempo real. Debido a que, para cada frame tiene que efectuar la detección de objetos buscados y dibujarlos, para ser ajustados en el video. Ejecutar este proceso en tiempo real, con una cantidad de fotogramas por segundo (FPS) hacen que la calidad del vídeo se vea afectada para esto se requiere algoritmos más rápidos (Torres Alonzo, 2020).

YOLO: la arquitectura Yolo (You Only Look Once) es un tipo de red R-CNN, que es utilizada para el reconocimiento de imágenes en tiempo real. Este algoritmo se distingue de los demás por ser extremadamente rápido y preciso. Como su nombre lo indica, solo requiere ver la imagen una sola vez, lo que hace considerablemente rápido en comparación con otros modelos. Además, permite realizar intercambios de velocidad y precisión a través del cambio de tamaño de la arquitectura, sin la necesidad de efectuar nuevos entrenamientos (Zavala Salas, 2020).

Estructura de la red Yolo: la arquitectura original está basada en la red neuronal convolucional GoogleNet, la misma se conforma de 24 capas convolucionales como se muestra en la Figura 19. Donde las capas iniciales de la red se encargan de extraer las características más significativas de la imagen, mientras que las capas completamente conectadas predicen las probabilidades de la salida y las coordenadas del objeto de estudio.

Figura 19

Estructura original de la red Yolo



Nota: Diagrama de la estructura original de la red Yolo. Tomado de *Evolution of Yolo Algorithm And Yolov5: The State-of-The-Art Object Detection Algorithm* [Imagen], Do Thuan, 2021, www.theseus.fi/bitstream/handle/10024/452552/Do_Thuan.pdf?isAllowed=y&sequence=2

> A diferencia de los modelos anteriores, que reutilizan los clasificadores para ejecutar la detección de objetos. Yolo encuadra el reconocimiento de objetos como un problema de regresión en cuadros delimitados, separados espacialmente y con las probabilidades de las clases asociadas. Este modelo es capaz de procesar imágenes en tiempo real, a una cantidad de 45 cuadros por segundo. (Sanz Cabreros, 2019, p.31)

En comparación a otros algoritmos de reconocimiento, que realizan la predicción de la clase a la que pertenece el objeto y la ubicación del mismo en la imagen. Yolov3, tiene la capacidad de reconocer o detectar múltiples objetos en una imagen, a través de una sola red neuronal (Gama García, 2020).

Yolov1: en la primera versión el algoritmo redimensionaba las imágenes a una resolución 448 x 448, y dividía la imagen en una cuadrícula de tamaño *S*x*S*, para la detección de objetos cada una de las celdas de la cuadrícula predice, cuadros delimitadores *B*. Cada predicción se compone de un vector de 5 dimensiones [*x, y, w, h,* puntuación de confianza]. Donde (*x,y*) representan la esquina superior de cuadro delimitador, *w* (alto), *h* (ancho). Además, el resultado de la regresión es de tamaño *S*x*S*x*B*x5, al mismo tiempo es capaz de predecir las probabilidades de clase para cada una de las celdas, lo que significa que la salida de la clasificación es de tamaño *S*x*S*x*C*, donde *C* es el número de clases (Do Thuan, 2021).

Yolov2: esta versión presentó mejoras en relación a Yolov1, debido a que se añadieron capas de normalización por lotes en todas las capas convolucionales, lo que permitió aumentar las capacidades de regularización de la red. Donde mejoró el mAP en más de 2%, además Yolov2 utiliza el modelo Darknet19 que está formado por 19 capas convolucionales, 5 capas máximas de agrupación y requiere menos operaciones. Esto hace que Yolov2 sea más rápido que Yolov1 (Do Thuan, 2021).

Yolov3: de la misma forma que las anteriores versiones, el modelo Yolov3 toma como entrada *B* cuadros de anclaje predefinidos, la pérdida de regresión es idéntica a

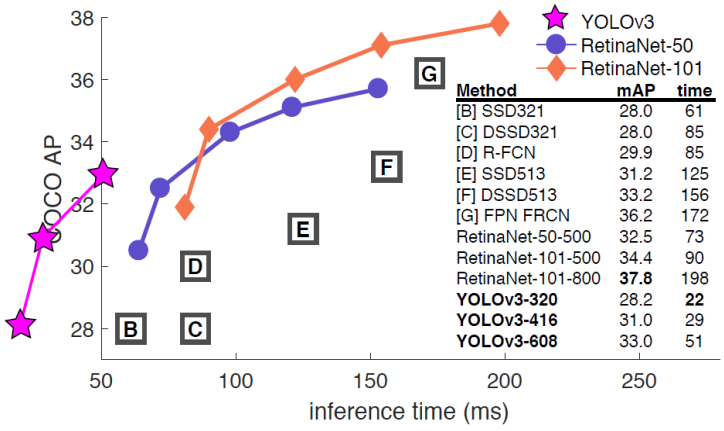
Yolov1, mientras que la regresión logística se utiliza para predecir la puntuación de confianza para cada cuadro delimitador.

> La arquitectura Yolov3, cuenta con un extractor de características hibrido Darknet-53 para el entrenamiento. Donde utiliza 53 capas convolucionales 3x3, 1x1, algunas conexiones de acceso directo y capas residuales. Además, tiene 53 capas convolucionales apiladas para la detección de objetos, convirtiéndolo así a Yolov3 en una arquitectura subyacente totalmente convolucional de 106 capas. (Zavala Salas, 2020, p.43)

En la Figura 20 se indica el rendimiento del algoritmo Yolov3 en comparación a otros algoritmos empleados en la detección de objetos.

Figura 20

Comparación de Yolov3 con otros algoritmos



Nota: Comparación del algoritmo Yolov3 con otros algoritmos. Tomado de *YOLOv3 se ejecuta mucho más rápido que otros métodos de detección con un rendimiento comparable utilizando una GPU M40 / Titan X*, [Imagen], Vidushi , 2021, www.viso.ai/deep-learning/yolov3-overview

Predicción de cuadros delimitadores: Cada cuadro delimitador en el Yolo original consta de cuatro predicciones: *x, y, w, h*. El centro de un cuadro estaba representado por coordenadas (*x,y*) relativas a los límites de la celda de la cuadrícula. El ancho w y la altura h se predicen en relación con toda la imagen. Para la segunda versión de Yolo el enfoque cambió debido a que se utilizó cuadros anclajes (anchors boxes) y offsets predichos en lugar de coordenadas. La predicción de offsets en lugar de coordenadas simplificó el problema y facilitó el aprendizaje de la red. El anclaje inicializa con dos dimensiones de anclaje anteriores: ancho *pw* y alto *ph* la red utiliza estas prioridades para predecir las coordenadas de altura *th*, ancho *tw* y centro (*tx,ty*) (Luuk Heinsius, 2021).

En la Figura 21 se indican de forma gráfica el problema de aprendizaje basado en el anclaje. Donde las siguientes ecuaciones ayudan a convertir las predicciones para obtener los cuadros delimitadores:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = \rho_w \, . \, e^{t_w}$$

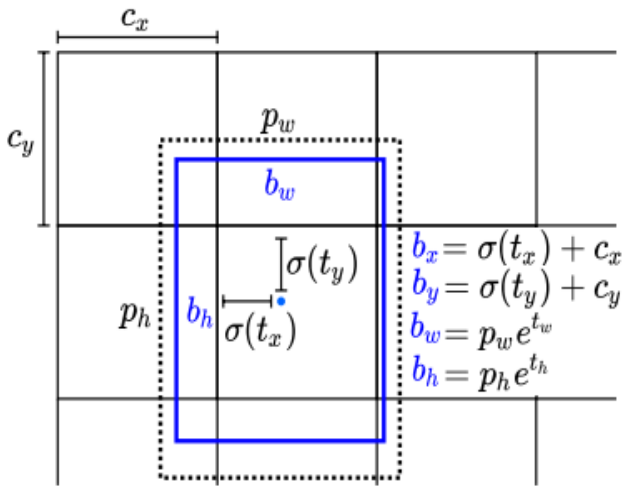
$$b_h = \rho_h \, . \, e^{t_h}$$

Los priores de la caja de anclaje o anchors boxes, están determinados por la agrupación de K-Means. El modelo Yolov3 emplea la regresión logística para predecir una puntuación de objeto para cada una de las cajas delimitadoras. De manera que, si el cuadro delimitador a la caja de anclaje no es la mejor, pero se superpone a un objeto por encima de un umbral (de un 0.5), se ignora la predicción (Ruz Gómez & Simón

Rodríguez, 2020).

Figura 21

Cajas delimitadoras con dimensiones previas y predicciones



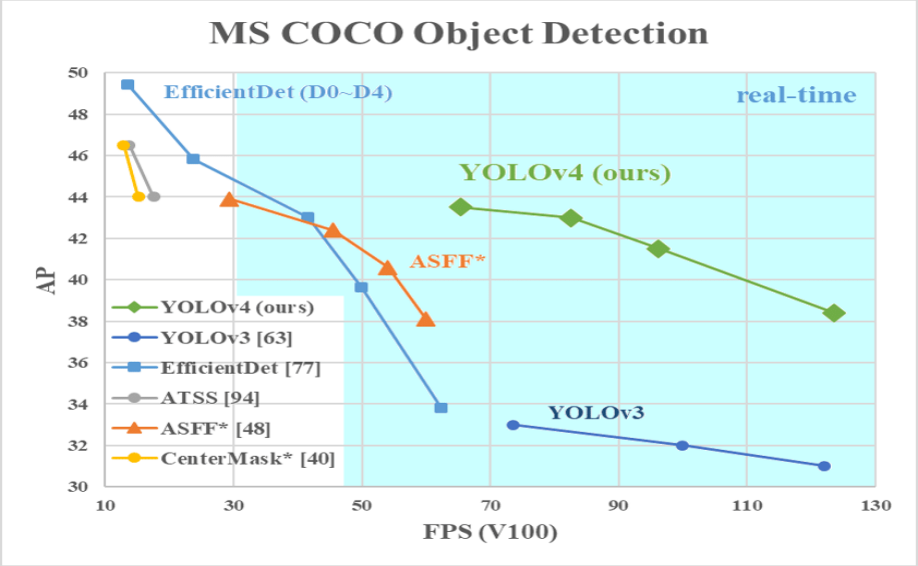
Nota: Cajas delimitadoras esta muestra la predicción de la altura, el ancho de la caja así como el offsets desde el centro de la caja. Tomada de *Real-Time YOLOv4 FPGA Design with Catapult High-Level Synthesis with Catapult High-Level Synthesis* [Imagen], Luuk Heinsius, 2021, www.essay.utwente.nl/86465/1/Heinsius_MA_EEMCS.pdf

Yolov4: esta versión del algoritmo presenta algunas mejoras considerables respecto a su predecesor Yolov3, en relación a la velocidad de interferencia y a su precisión en razón del 10 al 12% como se muestra en la Figura 22 (Ruz Gómez & Simón Rodríguez, 2020).

.

Figura 22

Comparación de la respuesta de Yolov4 con otros algoritmos



Nota: Diagrama de la comparación del Yolov4 con otros algoritmos para la detección de objetos. Tomada de *YOLOv4: velocidad y precisión óptimas en la detección de objetos* [Imagen], Mark Liao, 2020, www.researchgate.net/figure/Comparison-of-the-proposed-YOLOv4-and-other-state-of-the-art-object-detectors-YOLOv4_fig1_340883401

Estructura de la red Yolov4: de acuerdo con (Bo Gong, Daji Ergu, Ying Cai, & Bo Ma, 2020), esta arquitectura de red es más compleja que la de Yolov3, debido a que se compone de tres partes esenciales que son:

- El backbone: utiliza la red CSPDarknet53, que se basa en la estructura ResNet que es empleada para asegurar que la red tenga profundidad y al mismo tiempo pueda aliviar el gradiente de desaparición.
- El neck: la red Yolov4 emplea dos redes la primera es la SPP (Agrupación de Pirámides Espaciales), para aumentar eficazmente la receptividad y ayudar a separar las características contextuales. La segunda es la red PANet (Red de

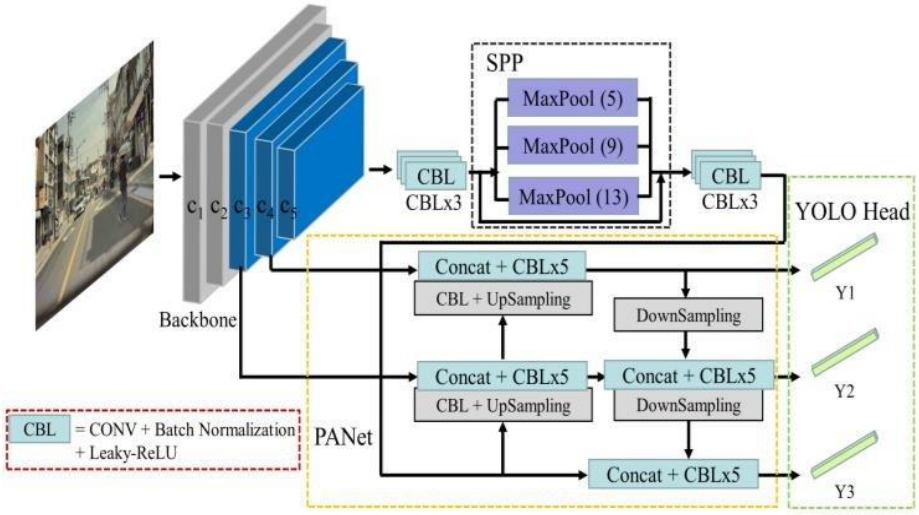
Agregación de Rutas) permite acortar el camino que conecta la información de bajo y alto nivel, y los parámetros convergentes a diferentes niveles.

- El header: la red Yolov4 hereda la estructura de la cabeza de Yolov3.

En la Figura 23 se indica la arquitectura de la red neuronal Yolov4.

Figura 23

Arquitectura de la red neuronal Yolov4



Nota: Arquitectura completa de la red Yolov4. Tomada de *ResearchGate* [Imagen], Hanxiang Wang, 2021, www.researchgate.net/figure/Overall-structure-of-YOLOv4-including-CSPDarknet-backbone-SPPnet-PANet-and-3-YOLO_fig2_344919620

Además, la arquitectura Yolo cuenta con una versión reducida que es la estructura Yolo-Tiny, está es la compresión de la arquitectura original, y fue diseñada específicamente para simplificar la estructura de la red. Con el propósito de reducir el tiempo de los parámetros empleados en detección de objetos y sea implementada en dispositivos que tengan limitados recursos computacionales como dispositivos móviles o SBC (Single Board Computer).

CAPÍTULO III: MARCO METODOLÓGICO

Descripción del Área de Estudio

La presente investigación, está orientada al sector agrícola del país, específicamente a los productores del guisante o arveja los que son afectados por la plaga thrips (*Kakothrips Robustus Uzel*) en sus cultivos. Esta investigación, se llevará a cabo en la provincia del Carchi, en el cantón Bolívar, específicamente en la parroquia de Los Andes. La parroquia cuenta con una población de 2.260 habitantes y una extensión territorial de 61.11 Km². Además, tiene una temperatura promedio que oscila entre 12 y 18°C, y una altitud promedio de 2550 m.s.n.m., asimismo tiene una densidad poblacional de 36.98 habitantes por Km² (GAD Municipal del Cantón Bolívar, s.f.).

Figura 24

Mapa del suelo para la agricultura de la parroquia Los Andes

Nota: Mapa de la capacidad de uso del suelo para la agricultura en la parroquia Los Andes. Tomado de *Actualización del Plan de Desarrollo y Ordenamiento de la Parroquia Los Andes 2015 – 2019* [Imagen], app.sni.gob.ec, 2016.

Enfoque y Tipo de Investigación

La presente investigación empleará un enfoque cualitativo y cuantitativo, el que será usado como técnica, para la recolección y el análisis de datos empíricos de las observaciones de campo que se realizará al objeto de estudio. De manera que, el sistema propuesto, proporcione respuesta a las interrogantes de la investigación y permita experimentar la hipótesis planteada con base en los resultados alcanzados, sin afectar de ninguna forma el objeto de estudio.

El presente trabajo utilizará la siguiente modalidad de investigación:

Investigación Bibliográfica

La investigación será bibliográfica, debido a que se utilizaran fuentes teóricas como artículos científicos, trabajos de grado y libros sobre temas relacionados con la presente investigación. Además, se usarán manuales sobre las herramientas y librerías del software que será empleado en el análisis y el procesamiento de imágenes, así como para la implementación y el entrenamiento de modelos y algoritmos basados en DL. De manera que, permita la abstracción de la información teórica y tecnológica de los diferentes trabajos de investigación.

Investigación Documental

Esta investigación en esencia, utilizará fundamentación teórica de artículos científicos y trabajos de grado relacionados a la propuesta planteada. Los que contribuirán

en la elaboración del marco teórico, además permitirán reutilizar algunos de los experimentos que fueron efectuados en estas investigaciones, con la finalidad de tratar que los resultados que se alcancen sean óptimos y apegados a la realidad.

Investigación de Campo

La investigación será de campo, ya que para el sistema propuesto se requerirá de la interacción de personas relacionadas al cultivo del guisante o arveja. Donde su criterio ayude a detectar de forma acertada las lesiones necróticas, que son causadas por la plaga thrips, en el cultivo y a través de la extracción de características significativas y etiquetado de las mismas en las imágenes. De manera que, permitan efectuar la corrección de errores en la ejecución de las pruebas de funcionamiento y en los resultados alcanzados en primera instancia, y así mejorar la precisión del sistema.

Investigación de Aplicada

Esta investigación será aplicada, debido a que se recolectarán imágenes de los cultivos que se encuentren sanos, infectados o con secuelas que fueron causadas por la plaga thrips. Las imágenes se almacenarán en un repositorio digital y serán procesadas por el sistema de reconocimiento de lesiones necróticas. De tal forma que, a través de la implementación de las técnicas y algoritmos de DL, realicen en tiempo real la detección temprana de la plaga thrips en el guisante o arveja, y además permitan conocer el estado actual de los cultivos.

Nivel o Tipo de Investigación

Descriptiva

La investigación que se utilizará es de tipo descriptivo, puesto que se observa y se describe el fenómeno, que en este caso será la presencia de lesiones necróticas en los cultivos del guisante o arveja, y así reconocer la relación existente entre la variable independiente y dependiente del problema.

Causal-Experimental

También, se utilizará una investigación causal-experimental, porque a través de la investigación se manipulará la variable independiente mediante la elaboración de pruebas de funcionamiento al sistema desarrollado. Y a través, de la comprobación de experimentaciones a la variable de estudio (que es el reconocimiento de lesiones necróticas), permitirá conocer el estado actual del guisante o arveja. De modo que, se obtenga evidencia en la relación causa y efecto del fenómeno de estudio.

Población y Muestra

La presente investigación será aplicada a una determinada población de los cultivos del guisante o arveja, en la etapa de desarrollo del fruto; debido a que el espacio de muestreo es muy extenso. Para esta muestra se utilizarán imágenes que fueron capturadas desde las parcelas agrícolas del cultivo, almacenadas en un repositorio digital y procesadas por el sistema desarrollado.

El tamaño de la población muestral, que se empleará en la presente investigación corresponde a 100 casos de cultivos, que se encuentren en buen estado y otros que presenten lesiones necróticas o tengan secuelas causadas por la plaga thrips. Para cada uno de los casos, se considerarán criterios de inclusión y exclusión, debido al estado actual que presente el guisante o arveja. En la Tabla 5 se indican las consideraciones que serán utilizadas para el tamaño de la población muestral.

Tabla 5

Población de estudio de la investigación

CULTIVO	POBLACIÓN	PATOLOGÍA	CONDICIONES
Guisante o Arveja	50	Cultivos que presentan lesiones necróticas causadas por la plaga thrips	Sin plaga thrips
	50		Con plaga thrips

Nota: En la tabla se muestra las consideradas de la población muestral para el análisis de la investigación. Realizado por el Autor.

Recolección de Información

Para la recolección de información, se realizarán algunas visitas a las parcelas agrícolas del cultivo. Donde se efectuarán capturas de imágenes al guisante o arveja, en la etapa de desarrollo del fruto (que tiene una duración aproximada de 35 días), con el uso de equipos digitales (como cámaras o smartphones).

En cambio, para la elección de las imágenes que serán utilizadas para el análisis del cultivo como en el entrenamiento de la red neuronal, es necesario contar con el apoyo de un experto en la producción del guisante o arveja. Debido a que, aportará con sus

conocimientos, al momento de efectuar un diagnóstico previo sobre la presencia de la plaga thrips en el cultivo, y así evidenciar el nivel de reconocimiento o detección del objeto de estudio, que tendrá el sistema desarrollado cuando se encuentre en funcionamiento.

Procesamiento y Análisis de Información

Una vez que se haya recolectado la información, desde las parcelas agrícolas del guisante o arveja, se realizará el análisis y procesamiento de las imágenes con la herramienta Roboflow (empleada para la creación de dataset), y con el uso de la red neuronal yolov4-tiny (técnica de DL para detección de objetos) se conseguirá el reconocimiento de lesiones necróticas provocadas por la plaga thrips en los cultivos.

Después de efectuar las pruebas de la detección de la plaga thrips en los cultivos, se realizará un análisis comparativo de los resultados obtenidos por el sistema, respecto al criterio del experto en la producción del cultivo (este se encargará de realizar un diagnóstico previo al cultivo a través de observaciones directas). De tal forma que, permitan constatar la veracidad y precisión del sistema desarrollado, y mediante la aplicación de un modelo estadístico validar la aceptación de la hipótesis planteada.

Consideraciones Éticas

Este trabajo de investigación será desarrollado de forma autónoma, equitativa, responsable y prudente. Debido a que, toda la información que se obtenga en todo el proceso de investigación será tratada de manera confidencial, y para uso exclusivo del

investigador bajo fines pertinentes, protegiendo la autonomía de las personas involucradas y de la información recopilada.

Análisis y Desarrollo del Sistema Propuesto

Análisis de Factibilidad

Factibilidad Técnica: desde el punto de vista técnico el proyecto es factible, gracias a que, los elementos de hardware (computador portátil, drone y SBC) y las herramientas de software (programas, librerías y recursos) están al alcance para el desarrollo del sistema propuesto. La propuesta esencialmente utilizará Python, que es un lenguaje de programación libre y multiplataforma orientada a objetos, y con la implementación de algunas librerías específicas, permitirán efectuar el análisis y procesamiento de imágenes. Además, para el entrenamiento de las técnicas y algoritmos de DL, en la detección del objeto de estudio.

Factibilidad Económica: el proyecto de investigación es económicamente factible, debido a que, todos los costos inmersos en el desarrollo del sistema serán asumidos en su totalidad por parte del investigador. De igual manera, al usar herramientas de software libre hace que también sea factible debido a que, no se requerirá de ningún tipo d**e** licenciamiento y podrá ser utilizado en cualquier PC o dispositivo SBC. Sin embargo, para el análisis se necesitará del asesoramiento del experto en la producción del cultivo.

Factibilidad Bibliográfica: el proyecto desde punto de vista bibliográfico es también factible puesto que, se cuenta con información concerniente al tema de investigación.

Donde se tomará como referencia diferentes fuentes de documentación como artículos científicos, trabajos de grado, revistas tecnológicas y publicación información (referente al tema) en Internet.

Análisis de la Arquitectura de la Red CNN Propuesta

Como se describió en el marco teórico, la arquitectura Yolo, es la más competente debido a los excelentes resultados que entrega en tiempo de respuesta como en precisión, lo que le ha permitido posicionarse por encima de otras arquitecturas que son utilizadas para la detección de objetos. Para el desarrollo de la propuesta planteada se utilizará la estructura Yolov4-Tiny.

La arquitectura Yolov4-Tiny, es aproximadamente ocho veces mayor que la versión original Yolov4 respecto a FPS (*Frames per Second*). Sin embargo, la precisión es de 2/3 que la de Yolov4 para un conjunto de datos MS Coco (AnalyticsVidhya, 2020). En cambio, que para la detección de objetos Yolov4-Tiny es la mejor opción, debido a la rapidez inferencia que alcanza en ambientes de trabajo en tiempo real.

Análisis del Software Requerido

Para el desarrollo del sistema propuesto, que permitan la detección temprana de la plaga thrips en los cultivos del guisante o arveja, se necesitará de algunos programas, herramientas y librerías de software libre. Además, de la aplicación de algunas técnicas y algoritmos basados en DL.

Lenguaje de Programación: el lenguaje de programación que se empleará en el desarrollo del sistema será Python. Esta es una herramienta libre que tiene una estructura de datos de alto nivel, y un enfoque simple a la programación. Debido a que, su sintaxis es ideal para el desarrollo de aplicaciones en áreas del aprendizaje automático en virtud de la amplia biblioteca que posee y a la facilidad de trabajo que tiene para el arreglo de datos.

Python permitirá la ejecución de algunas etapas, como el entrenamiento de la red neuronal yolov4-tiny, para detección temprana de la plaga thrips en los cultivos. Al utilizar Python, ligada a ciertas librerías de visión artificial como OpenCV, y otras de aprendizaje automático como TensorFlow, hacen que se convierta en el instrumento fundamental para el desarrollo del sistema propuesto.

En la Tabla 6, se muestran las librerías que serán utilizadas en el desarrollo del sistema propuesto.

Tabla 6

Librerías utilizadas en el desarrollo del sistema

LENGUAJE DE PROGRAMACIÓN	LIBRERÍA	FUNCIONALIDAD
	Cython	Módulos de extensión para Python en C y C++.
	H5py	Almacena grandes cantidades de datos numéricos.
	Imgaug	Manipula imágenes para aprendizaje automático.
	IPython	Shell de comandos para computación interactiva en múltiples lenguajes de programación.
	Keras	Diseñada para la experimentación con las redes de aprendizaje profundo.
	Matplotlib	Generación de gráficos a partir de datos contenidos en listas o arrays en Python y su extensión NumPy.
	NumPy	Funcionalidades matemáticas avanzadas para el manejo de vectores y matrices.
	Os	Interactúa con el sistema operativo subyacente en el que se ejecuta Python.
	OpenCV	Análisis de imágenes y aprendizaje automático desarrollada por Intel.

Python	Pandas	Análisis y manejo de datos ofreciendo estructuras de datos y operaciones en tablas numéricas.
	Pillow	Soporte para abrir, manipular y guardar diversos formatos de archivo de imagen diferentes.
	SciPy	Manejo de algoritmos y herramientas matemáticas, que contiene módulos para tareas como optimización, algebra lineal, funciones especiales, procesamiento de señales y de imágenes.
	Skimage	Procesamiento de imágenes que tiene algoritmos para segmentación, transformaciones geométricas, manipulación del espacio de color, análisis, filtrado, morfología, detección de características, etc.
	TensorFlow	Aprendizaje automático desarrollado por Google para la construcción y el entrenamiento de redes neuronales.
	Time	Provee de funciones para trabajar con tiempo y convertir entre representaciones.

Nota: En la tabla se muestran las librerías que se utilizarán en el desarrollo del sistema y la funcionalidad de cada una de ellas. Realizado por el Autor.

Aplicación Web: en el desarrollo de la aplicación web se empleará Steamlit, que es una librería Open Source de Python, utilizada en la ciencia de datos y en el aprendizaje automático. Esta librería permite crear aplicaciones interactivas y de alto rendimiento de manera rápida. Además, tiene la capacidad de actualizar automáticamente las aplicaciones web cada vez que el código fuente cambie o se modifique.

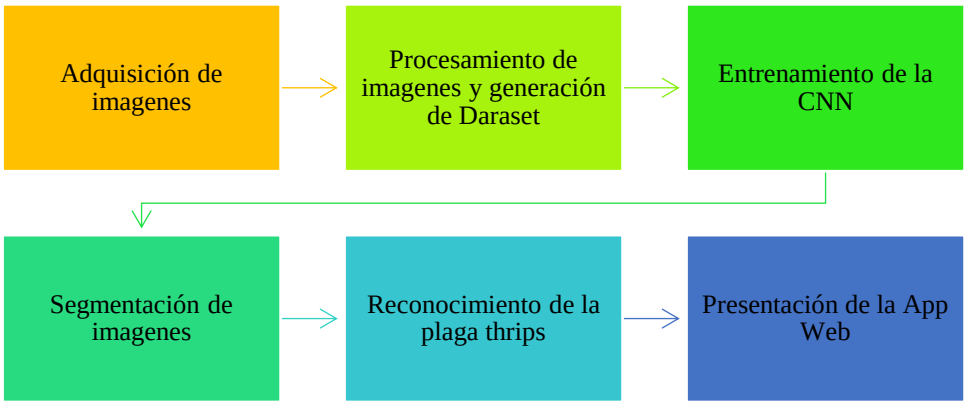
Diseño del Sistema Propuesto

El sistema que será desarrollado para el reconocimiento de lesiones necróticas en el guisante o arveja, está orientado a la detección temprana de la plaga thrips en el guisante o arveja, además permitirá conocer el estado actual del cultivo. Siendo la identificación de la plaga thrips la principal tarea que el sistema deberá cumplir. Donde la emisión y veracidad, de los resultados que se alcancen ayuden a que, los productores puedan establecer un tratamiento óptimo para el cultivo, y no exista la pérdida total del mismo.

El diseño del sistema propuesto, estará basado en la implementación de las técnicas y algoritmos de aprendizaje automático que es una parte esencial de la IA. En la Figura 25 se muestran las fases que serán parte en el desarrollo del sistema.

Figura 25

Diagrama de las fases para el desarrollo del sistema propuesto



Nota: Estructura de las diferentes etapas del proyecto de investigación. Realizado por el Autor.

A continuación, se describen cada una de las fases que constituyen el desarrollo del sistema propuesto.

Primera Fase: Adquisición de Imágenes

La fase de adquisición de imágenes es la parte fundamental para el desarrollo del sistema. Debido a que, depende en gran medida de la calidad de las imágenes y cómo estas se presentan. Por ejemplo, si las lesiones necróticas que fueron causadas por la plaga thrips en el cultivo son visibles o no. Las imágenes con las que se trabajó fueron

capturadas desde las parcelas agrícolas como se indica en la Figura 26, las mismas serán guardadas en formato JPG en un repositorio digital.

Figura 26

Guisante o arveja con lesiones necróticas

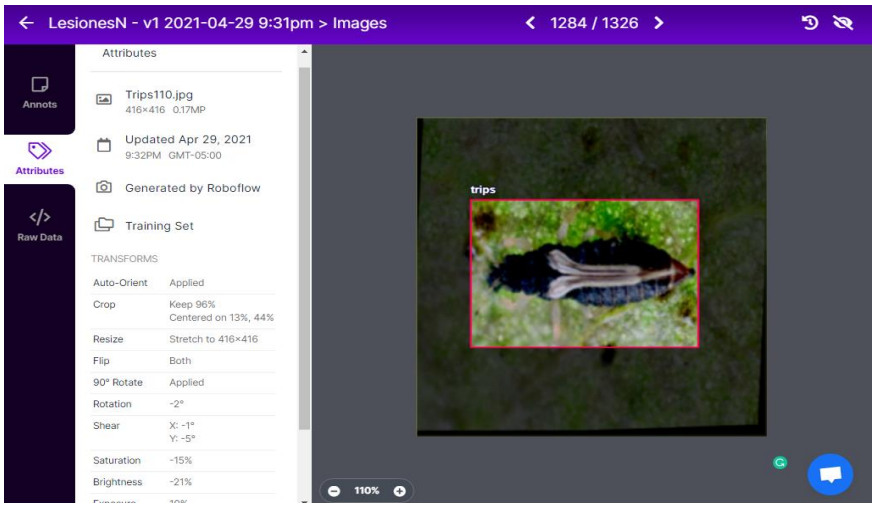
Nota: Lesiones necróticas causadas por la plaga thrips en los cultivos del guisante o arveja. Realizado por el Autor.

Segunda Fase: Procesamiento y Generación del Dataset

En esta fase se trabajó directamente, sobre las imágenes del cultivo que fueron guardadas en formato JPG. En las que se identificaron y etiquetaron, las características significativas del objeto de estudio en cada una de las imágenes con el uso de Roboflow. En esta sección también, se aplicaron filtros y atributos que permitieron mejorar la visualización y la detección de la plaga, y con base al reconocimiento de las características significativas se delimitó el área de las lesiones necróticas en el cultivo. De esta manera, se obtendrá un archivo ".xlsx" de las coordenadas donde se encuentra el objeto de estudio como se indica en la Figura 27.

Figura 27

Procesamiento y etiquetación de las imágenes



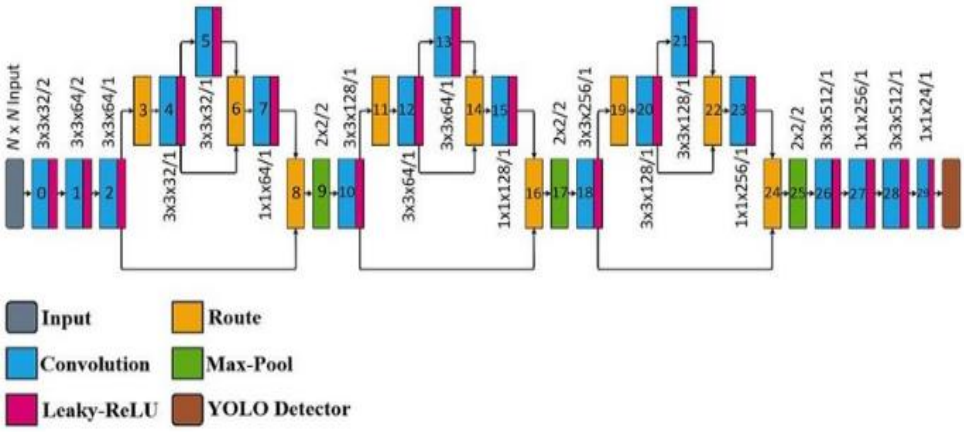
Nota: Procesamiento y etiquetación de las imágenes que serán parte del Dataset con el uso de la herramienta Roboflow. Realizado por el Autor.

Tercera Fase: Entrenamiento de la CNN

Para el entrenamiento de la red neuronal se utilizó la arquitectura yolov4-tiny, debido a que, posee un enfoque híbrido para la extracción de características gracias a que utiliza Darknet-29. Esta red neuronal se compone de 29 capas convolucionales sucesivas 3x3, 1x1 y de algunas capas de interconexión como muestra en la Figura 28.

Figura 28

Arquitectura de la red neuronal yolov4-tiny

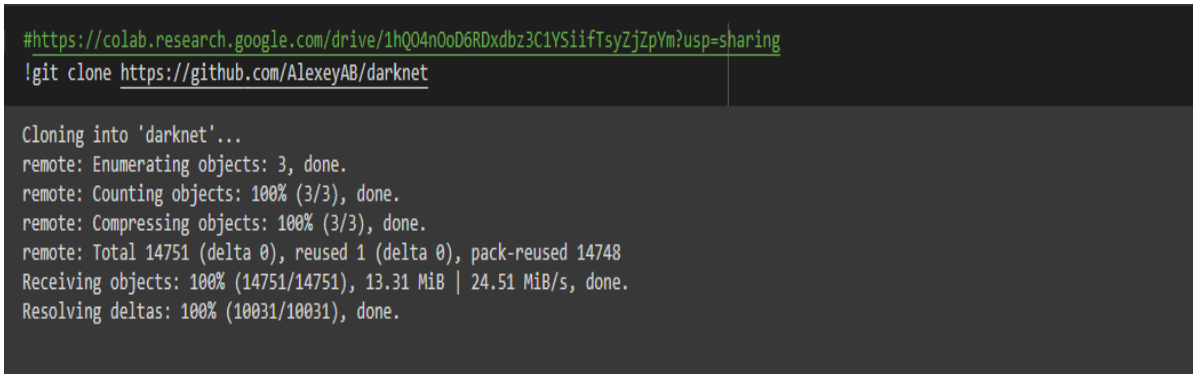


Nota: Diagrama de la arquitectura de la red neuronal yolov4-tiny.conv.29. Tomado de *ResearchGate* [Imagen], Pérez Montalbo, 2021, www.researchgate.net/figure/The-YOLOv4-Tiny-feature-extractor-and-backbone-for-the-MRI-brain-tumor-detection_fig3_348176421

El entrenamiento de la red yolov4-tiny, inicia con la instalación o clonación de la librería AlexeyAB/darknet, desde los repositorios de GitHub al ambiente de entrenamiento. Todo el entrenamiento de la red neuronal fue realizado en una máquina virtual de Google Colab. Debido a las prestaciones que este tipo de plataforma ofrece a nivel de GPU y de memoria RAM. En la Figura 29 de muestra la clonación de la librería AlexeyAB/darknet en la máquina virtual en el anexo A se encuentra en más detalle la clonación de la librería.

Figura 29

Clonación de la librería darknet para el entrenamiento

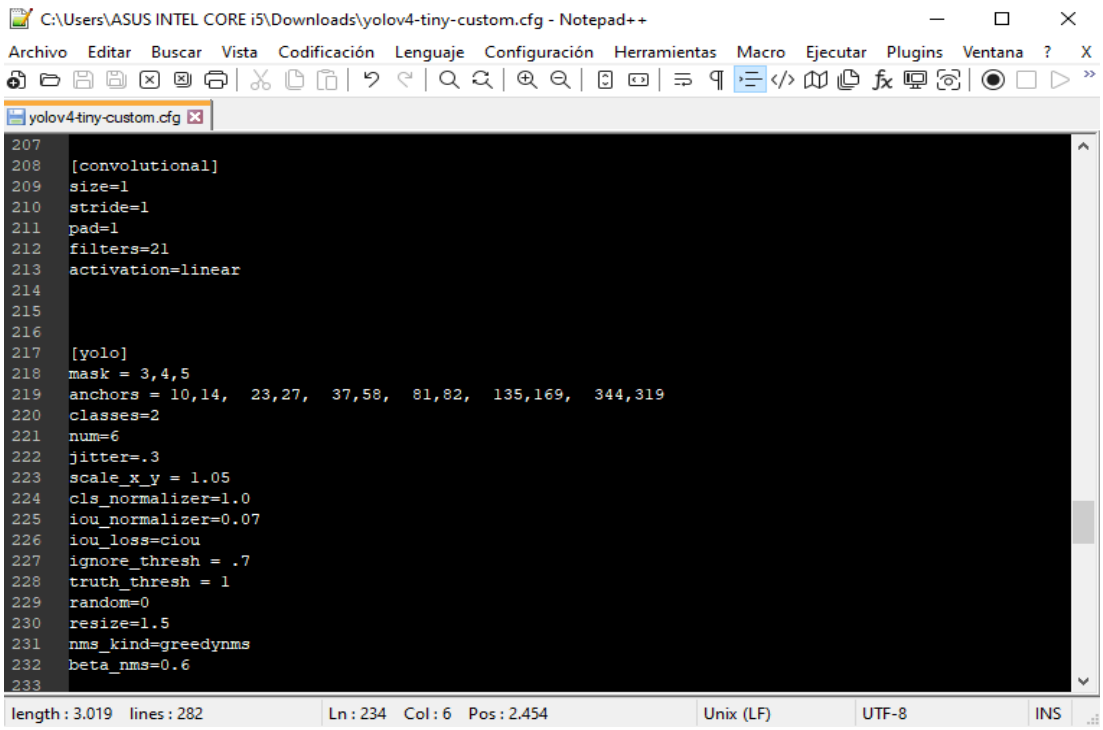


Nota: Clonación de la librería AlexeyAB/darknet para entrenamiento de la red yolov4-tiny en la máquina virtual de Google Colab. Realizado por el Autor.

Una vez descargadas las librerías, se requiere de la integración del dataset y de los archivos de entrenamiento (obj.data, obj.names, train.txt, valid.txt, yolov4-tiny-custom.cfg) los que tienen la información de cada una de las imágenes que serán utilizadas en el entrenamiento y en las pruebas de funcionamiento de la red neuronal yolov4-tiny. En el anexo B se indica todo el proceso de la creación y configuración de los archivos de entrenamiento. En la Figura 30 se observa la configuración del archivo yolov4-tiny-custom.cfg, el que contiene todos los parámetros que serán empleados en el entrenamiento de la red como el número de épocas, clases, filtros y grado de confianza para la detección de objetos.

Figura 30

Configuración del archivo yolov4-tiny-custom.cfg



Nota: Configuración del archivo yolov4-tiny-custom.cfg para el entrenamiento de la red neuronal. Realizado por el Autor.

Después, se ejecuta la integración del dataset a la máquina virtual de Google Colab en el directorio darknet/data como se indica en la Figura 31.

Figura 31

Integración del dataset al directorio darknet/data

```
# Descomprimir el dataset y su contenido en la carpeta /darknet/data/
!unzip ../obj.zip -d data/
```

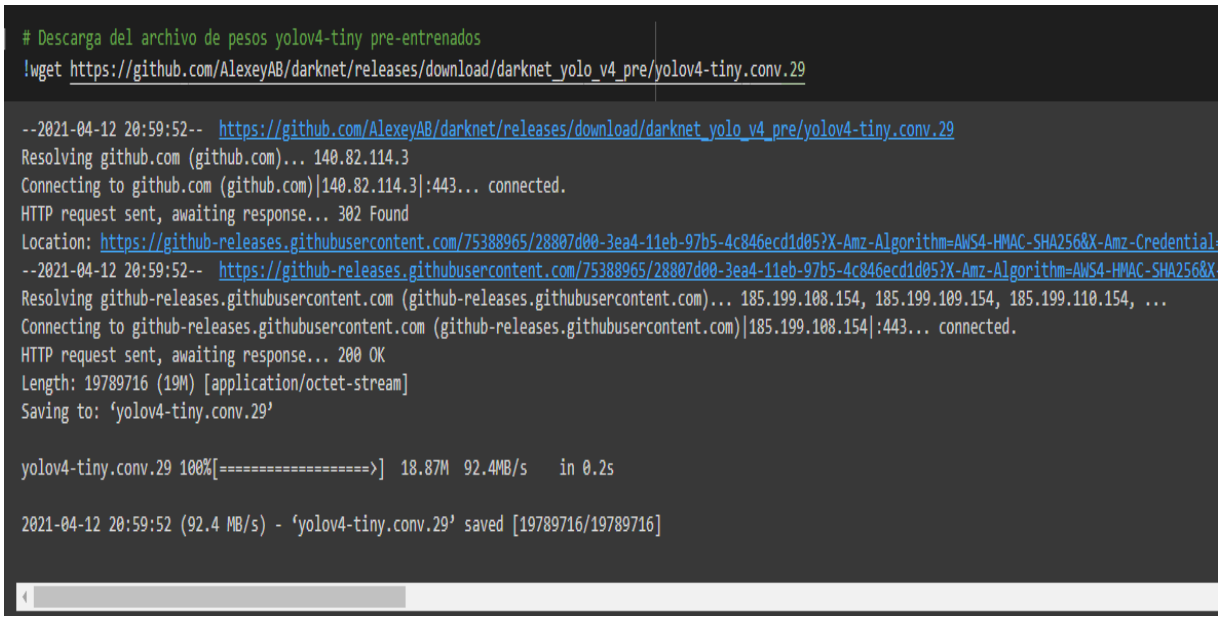
```
Archive:  ../obj.zip
   creating: data/obj/
  inflating: data/obj/Trips000_jpg.rf.0c4823a02e4909d81977bc4b1f20efaa.jpg
  inflating: data/obj/Trips000_jpg.rf.0c4823a02e4909d81977bc4b1f20efaa.txt
  inflating: data/obj/Trips000_jpg.rf.ca0c3a8a160695d808525158b7996a4d.jpg
  inflating: data/obj/Trips000_jpg.rf.ca0c3a8a160695d808525158b7996a4d.txt
  inflating: data/obj/Trips000_jpg.rf.edaf71961fdb7f7d2f2092e91b126f7a.jpg
  inflating: data/obj/Trips000_jpg.rf.edaf71961fdb7f7d2f2092e91b126f7a.txt
  inflating: data/obj/Trips001_jpg.rf.932a68d970ec47d04290120f6b960ce8.jpg
  inflating: data/obj/Trips001_jpg.rf.932a68d970ec47d04290120f6b960ce8.txt
  inflating: data/obj/Trips002_jpg.rf.d2244fd3ebad9b92360d6eed13840ea0.jpg
  inflating: data/obj/Trips002_jpg.rf.d2244fd3ebad9b92360d6eed13840ea0.txt
  inflating: data/obj/Trips003_jpg.rf.6638bfa989d24777896c3bc51994cb2d.jpg
  inflating: data/obj/Trips003_jpg.rf.6638bfa989d24777896c3bc51994cb2d.txt
  inflating: data/obj/Trips004_jpg.rf.0eb192618383cbbbc5a01bee9cb11596.jpg
  inflating: data/obj/Trips004_jpg.rf.0eb192618383cbbbc5a01bee9cb11596.txt
  inflating: data/obj/Trips004_jpg.rf.3877444de0565ec68b074078777c978b.jpg
  inflating: data/obj/Trips004_jpg.rf.3877444de0565ec68b074078777c978b.txt
  inflating: data/obj/Trips004_jpg.rf.f01173fbb5e5ac532dc01f1233006e4d.jpg
  inflating: data/obj/Trips004_jpg.rf.f01173fbb5e5ac532dc01f1233006e4d.txt
  inflating: data/obj/Trips005_jpg.rf.ccd6146462d0b733a4da31b11ac63022.jpg
  inflating: data/obj/Trips005_jpg.rf.ccd6146462d0b733a4da31b11ac63022.txt
  inflating: data/obj/Trips005_jpg.rf.dbea3a7a235ca6f733c6b6c613fa651f.jpg
  inflating: data/obj/Trips005_jpg.rf.dbea3a7a235ca6f733c6b6c613fa651f.txt
  inflating: data/obj/Trips005_jpg.rf.f289469f8d2df05de8370d9fa2ecbd67.jpg
  inflating: data/obj/Trips005_jpg.rf.f289469f8d2df05de8370d9fa2ecbd67.txt
  inflating: data/obj/Trips006_jpg.rf.32baf7381dde010808d0bff368a784e3.jpg
  inflating: data/obj/Trips006_jpg.rf.32baf7381dde010808d0bff368a784e3.txt
```

*Nota***:** Integración de las imágenes que serán parte del entrenamiento de la red neuronal a Google Colab en el directorio darknet/data. Realizado por el Autor.

A continuación, se realiza la descarga de la arquitectura yolov4-tiny desde los repositorios de GitHub, la que contiene los pesos preentrenados de dicha estructura los mismos que serán integrados al entrenamiento de la red neuronal. En la Figura 32 se muestra la implementación de la librería darknet_yolo_v4_pre/yolov4-tiny.conv.29, en el anexo A se detalla la implementación de la librería en Google Colab.

Figura 32

Implementación de la arquitectura yolov4-tiny.con.29

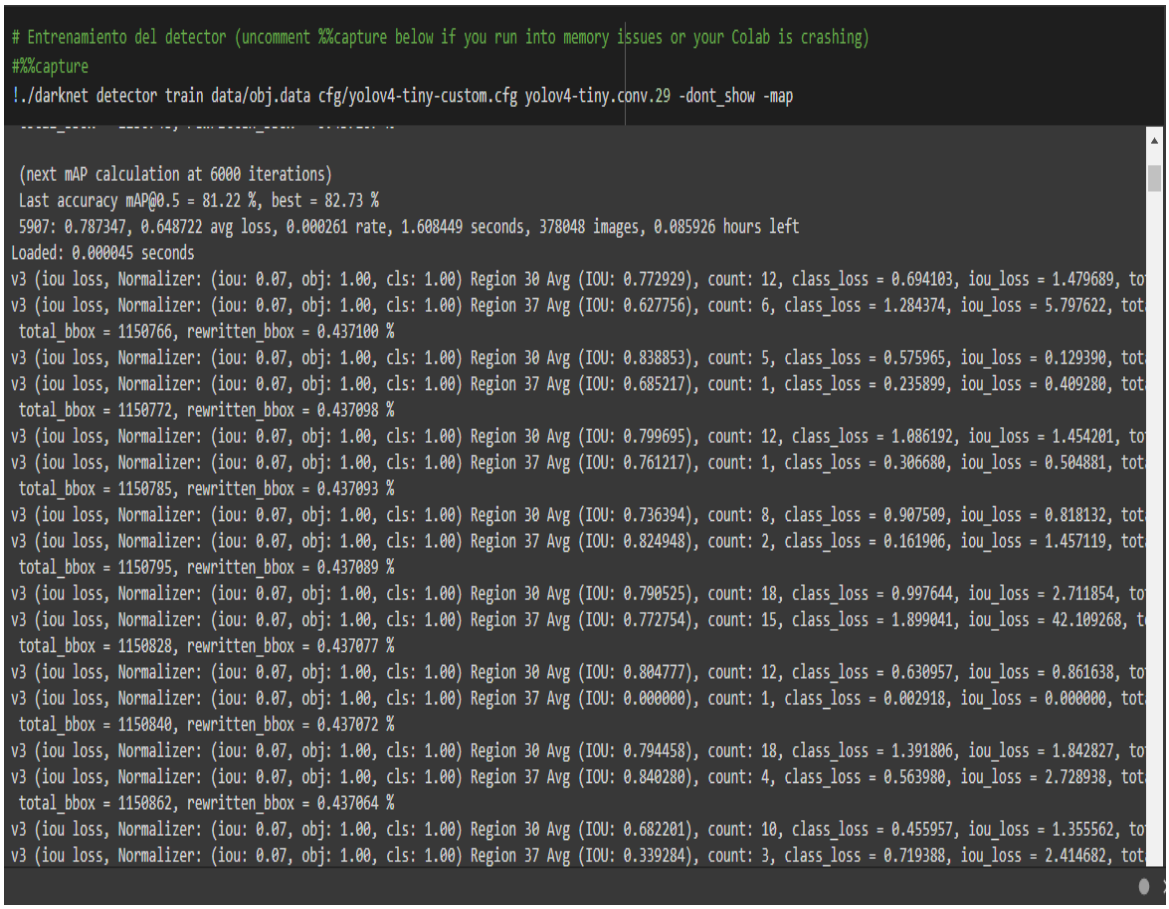


Nota: Descarga de la red neuronal yolov4-tiny.con.29 desde los repositorios de GitHub a la máquina virtual. Realizado por el Autor.

Finalmente, se ejecuta el entrenamiento donde se utilizó un dataset de 500 imágenes para 6000 épocas con ciertas variaciones, rotaciones y transformaciones que fueron recreadas por OpenCV, con la finalidad de ofrecer variedad al entrenamiento de manera que permita identificar de forma precisa la presencia de la plaga thrips en el guisante o arveja. En la Figura 33 se indica el entrenamiento de la red neuronal yolov4-tiny.conv.29.

Figura 33

Entrenamiento de la red neuronal yolov4-tiny.conv.29

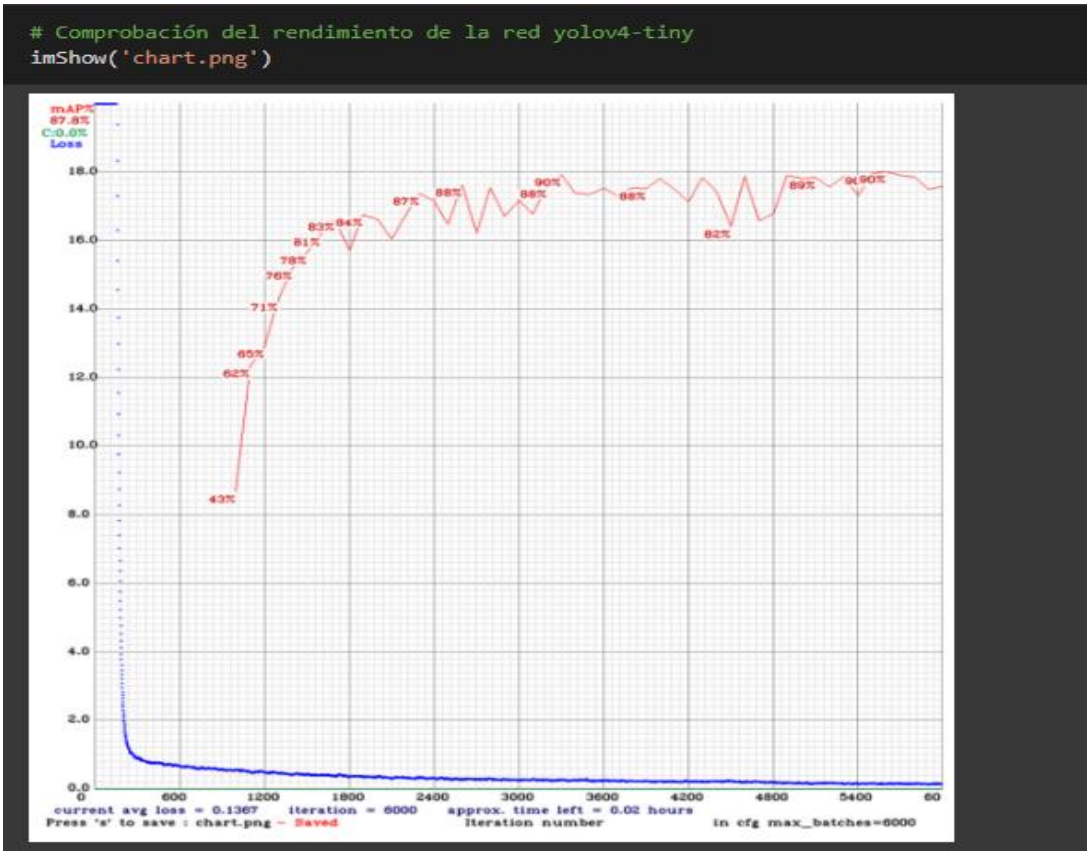


Nota: Entrenamiento de la red neuronal yolov4-tiny.conv.29 desde la máquina virtual de Google Colab. Realizado por el Autor.

En la Figura 34 se indica en valor promedio del rendimiento que alcanza la red yolov4-tiny.conv.29, cuando se realiza la detección del objeto de estudio en tiempo real. En el anexo A se especifican todos los pasos empleados para el entrenamiento, así como para la comprobación del rendimiento

Figura 34

Rendimiento de la red neuronal yolov4-tiny.conv.29



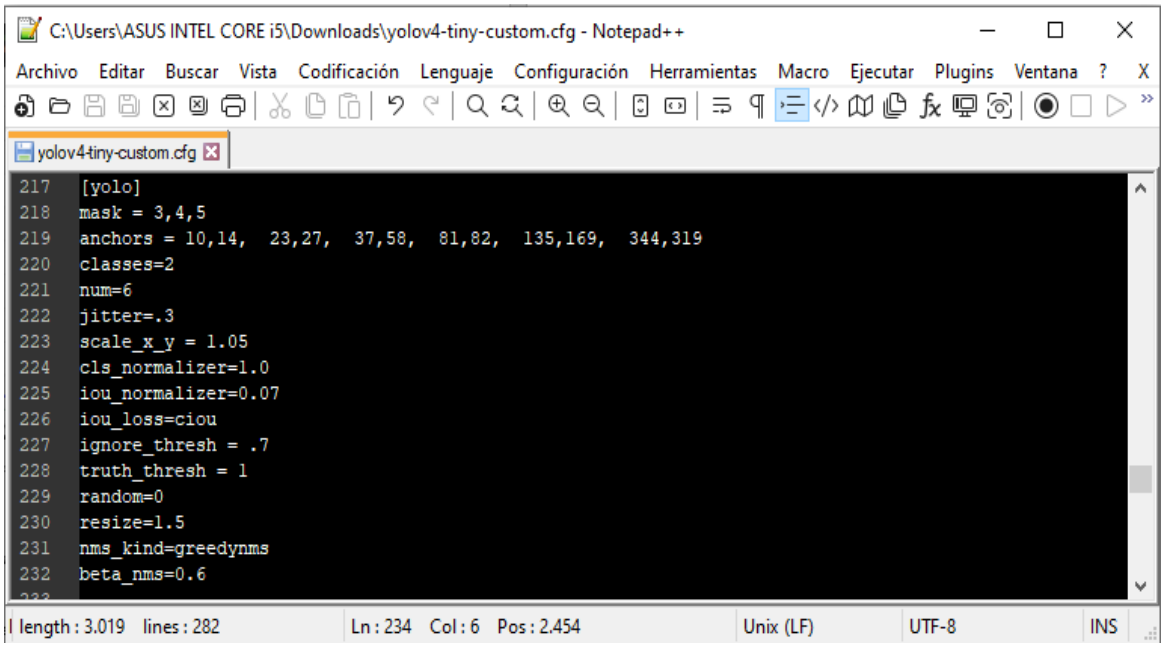
Nota: Comprobación del rendimiento que tiene la red neuronal yolov4-tiny en la detección de plaga thrips. Realizado por el Autor.

Cuarta Fase: Segmentación de Imágenes

En esta fase se realizará, la delimitación de los objetos que fueron detectados en cada una de las imágenes. Para esta tarea se realizará, un escaneo a la imagen donde se identificarán las regiones en las que se encuentran él o los objetos de estudio. A este tipo de regiones se las conoce como anchors box (no son más que cuadros de diferentes tamaños), los que se distribuyen por toda la imagen. En la Figura 35 se indican los diferentes anchors box que serán utilizados por la red yolov4-tiny.

Figura 35

Anchors box en la red neuronal yolov4-tiny.conv.29



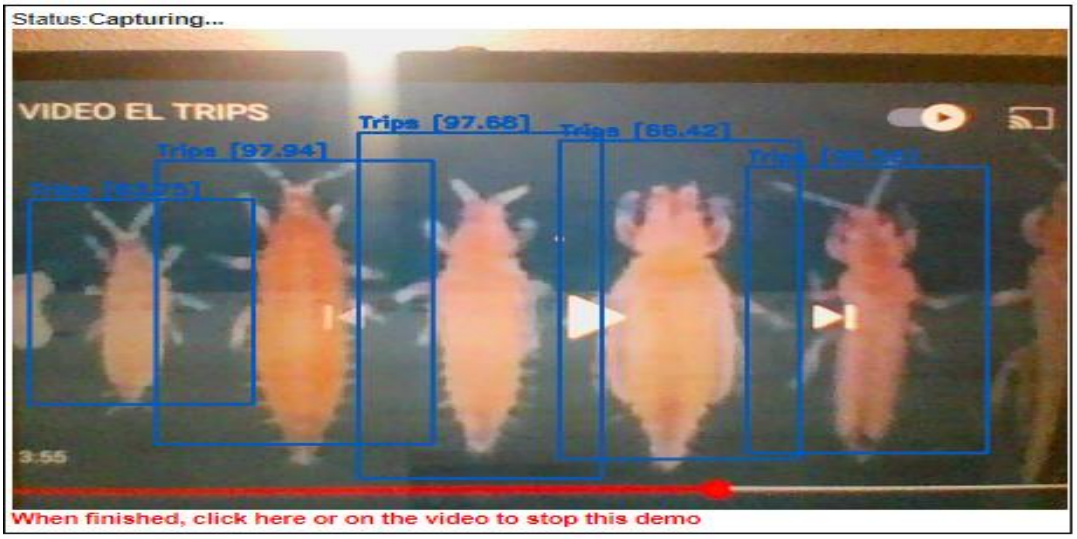
Nota: Configuración de los anchors box utilizados por la red neuronal yolov4.tiny para la detección de objetos. Realizado por el Autor.

Para la segmentación de imágenes la red yolov4-tiny.conv.29, efectuará miles de predicciones, donde presentará solo aquellas que son objetos. Esta tarea se encarga de eliminar la mayor cantidad de cuadros, a través del proceso de supresión máxima para obtener el resultado de interferencia única, en función de la probabilidad de la clase del objeto.

De manera que, si el anchors box en el que se encuentra el objeto de estudio no está centrado, la red yolov4-tiny.conv.29 ejecuta varios ajustes a estos cuadros, a través de predicciones. Donde calcula y muestra él o los objetos de la imagen que tienen el cuadro delimitador con mayor valor de IoU, como se observa en la Figura 36.

Figura 36

Segmentación de objetos por la red neuronal yolov4-tiny



Nota: Resultado de la segmentación de los objetos detectados por la red yolov4-tiny desde Google Colab. Realizado por el Autor.

Quinta Fase: Detección de la Plaga Thrips

En esta fase se presentarán los resultados que se obtuvieron en la detección del objeto de estudio. Esta inicia con la elección del archivo o imagen que será utilizada para el reconocimiento de lesiones necróticas causadas por la plaga thrips en el guisante o arveja. A continuación, se realizará la comprobación en tiempo real sobre la presencia de la plaga thrips en los cultivos. Finalmente, se efectuará un pronóstico acerca del estado actual de los cultivos. En la Figura 37 se indica la detección de la plaga thrips en los cultivos del guisante o arveja en imágenes, en el anexo C se encuentra en más detalle el proceso de la detección

Figura 37

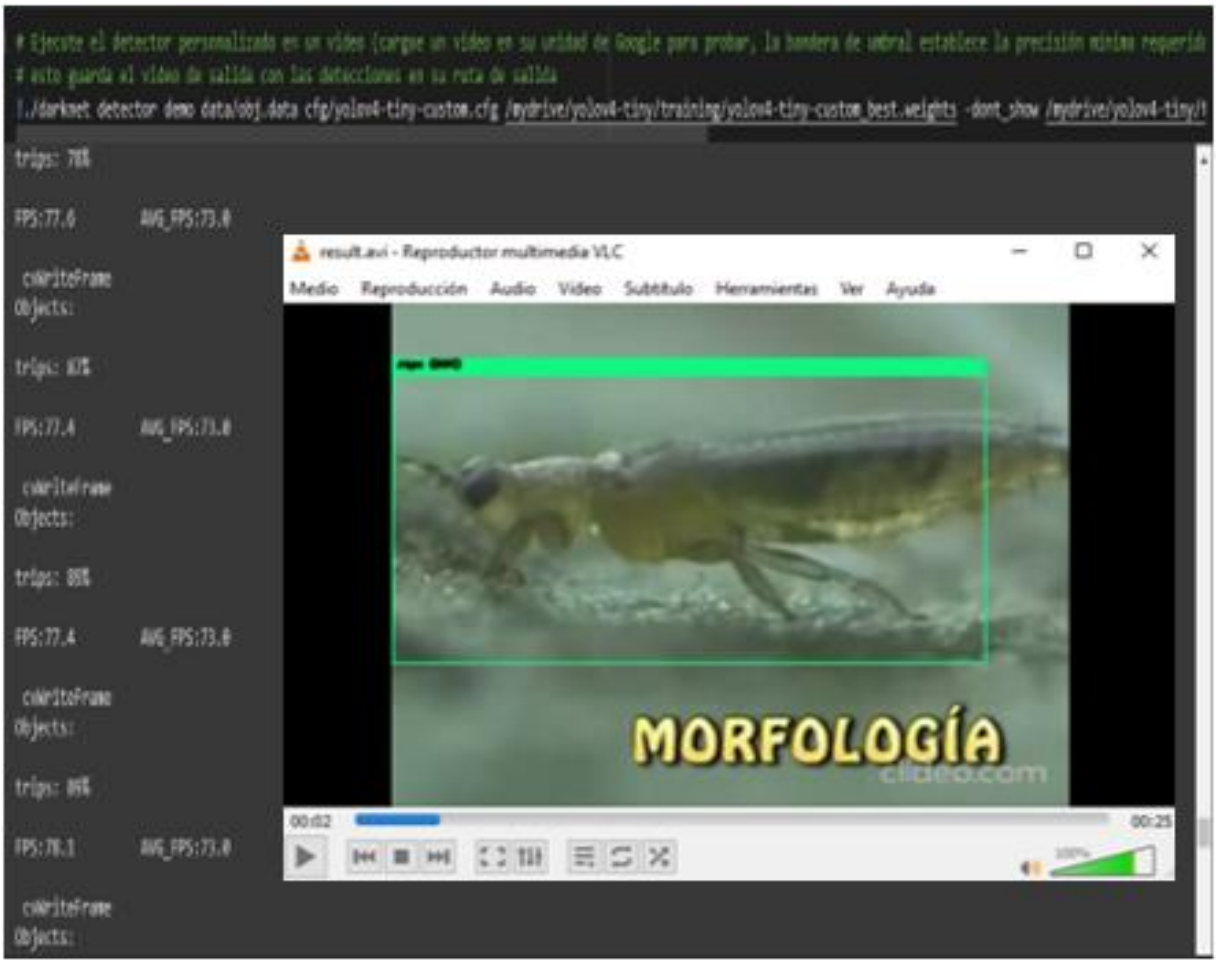
Detección de la plaga thrips en el guisante o arveja en imágenes

Nota: Detección de la plaga thrips en el guisante o arveja en imágenes desde Google Colab. Realizado por el Autor.

En esta sección se observó que la detección del objeto de estudio es muy efectiva. Debido a que, los resultados alcanzados en la detección temprana de la plaga thrips en los cultivos tuvieron una precisión del 98%. Además, se pudo evidenciar que fueron también positivos al realizar la detección de la plaga thrips sobre un video, gracias a que se consiguió una predicción del 89%. El la Figura 38 se observa la detección de la plaga thrips en el guisante o arveja en un video, en el anexo C se encuentran los pasos utilizados en la ejecución de la tarea.

Figura 38

Detección de la plaga thrips en el guisante o arveja en video

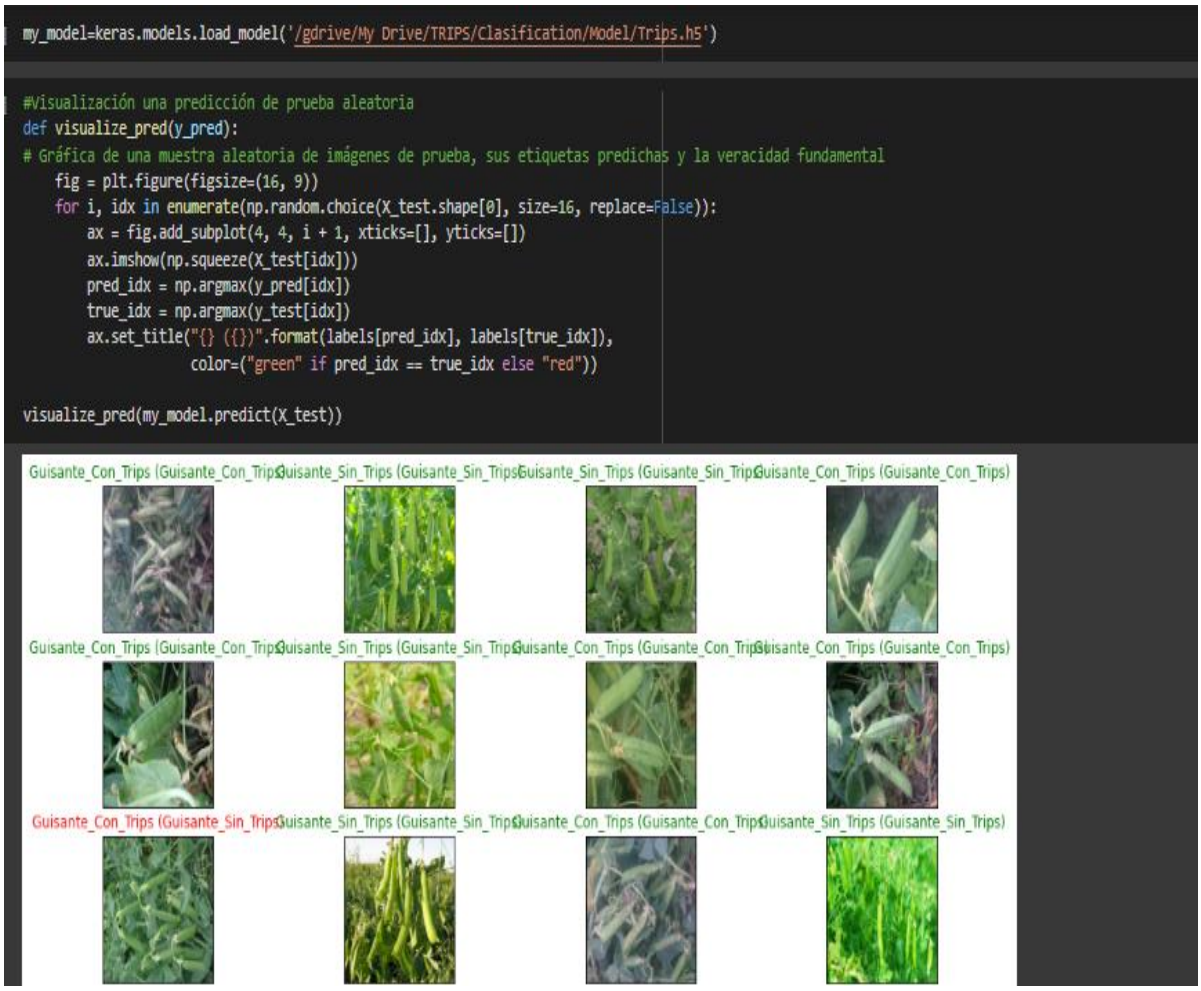


Nota: Detección de la plaga thrips en el guisante o arveja en un video desde Google Colab. Realizado por el Autor.

De igual manera, se corroboró que la predicción de las diferentes etiquetas y la autenticidad fundamental de las mismas fueron acertadas, después de la ejecución de una prueba aleatoria de las imágenes del guisante o arveja. Gracias a que, consiguió pronosticar de forma correcta el estado actual del cultivo como se observa en la Figura 39.

Figura 39

Predicción del estado actual del guisante o arveja.



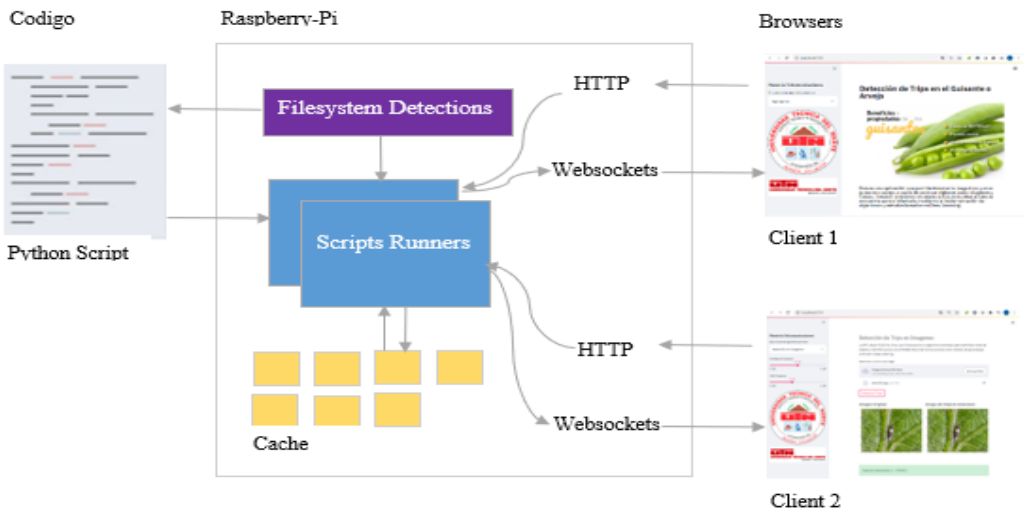
Nota: Predicción del estado actual del guisante o arveja desde Google Colab utilizada para el entrenamiento de la red neuronal. Realizado por el Autor.

Sexta Fase: Desarrollo de la Aplicación Web

En esta fase se detalla el desarrollo de la App Web, esta será una interfaz amigable e interactiva con el usuario final. Debido a que, permitirá la ejecución del funcionamiento total del sistema. En la Figura 40 se muestra la estructura de la App Web.

Figura 40

Estructura de la App Web desarrollada



Nota: Diagrama de bloques de la estructura de la App Web desarrollada para la deteccion de la plaga thrips. Realizado por el Autor.

A continuación, se detalla el proceso realizado en el desarrollo de la App Web:

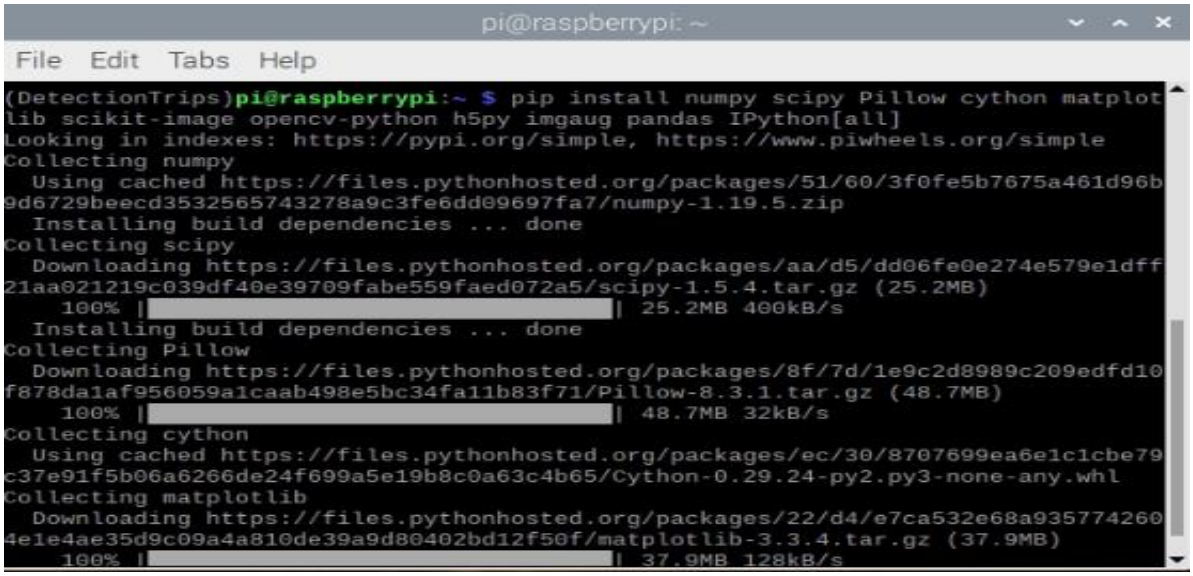
1. Análisis de la arquitectura yolov4-tiny, en este paso se determinan los requerimientos del sistema. Donde se definieron las herramientas, programas y librerías que permitirán la creación de scripts de código, así como los parámetros que utilizará la interfaz.
2. Instalación del software para el desarrollo de la App Web, a continuación, se especifican los pasos:

    - Primero se instaló el paquete anaconda que es una distribución libre y de código abierto para el lenguaje de programación Python, la que es empleada para la ciencia de datos y en el aprendizaje automático.

- Una vez instalado el paquete Python, se procede a la implementación de las diferentes librerías las que se detallaron en la sección del análisis del software requerido. En la Figura 41 se indica la instalación de las librerías utilizadas para el desarrollo de la App Web, en el anexo D se encuentra en detalle todo el proceso de instalación.

Figura 41

Instalación de las librerías para el desarrollo de la App Web



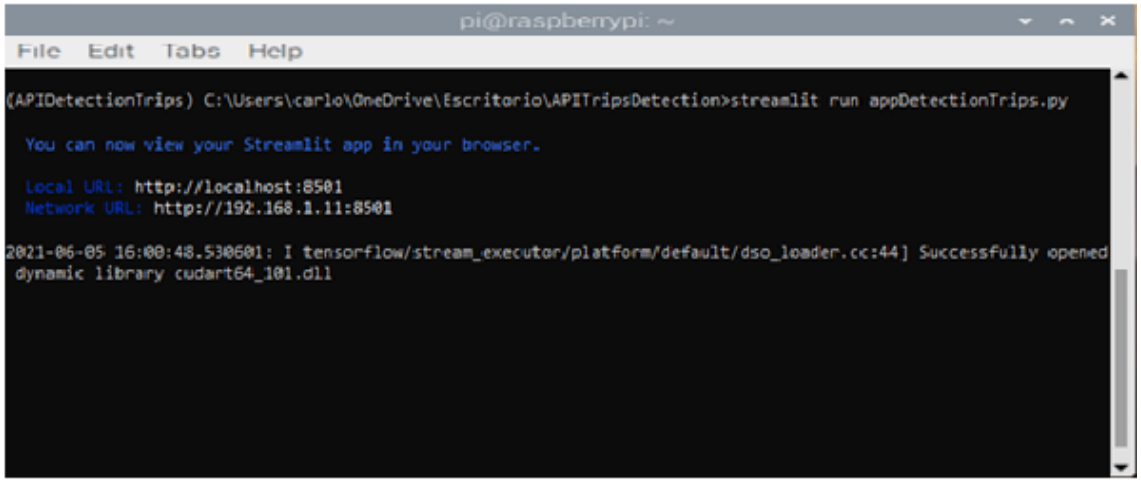
Nota: Instalación las librerías utilizadas para el desarrollo de la App Web de sistema de detección de la plaga thrips. Realizado por el Autor.

3. Generación de la estructura Streamlit, esta se encargará de las conexiones entre los scripts de código y los archivos del sistema de detección, además a los scripts de ejecución, y de estos últimos a los browsers del ordenador.
4. Creación de la estructura Streamlit, en este paso se realizó la creación del archivo del sistema de detección appDetectionThrips, como se indica en la Figura 42. Este contiene los diferentes parámetros que serán utilizados por la App Web. De igual

manera, se crearon los scripts de códigos y de ejecución; detection_imagenes.py, detection_video.py y estado_cultivo.py.

Figura 42

Inicialización del script appDetectionThrips



Nota: Inicialización del script appDetectionThrips para desarrollo de la APP Web. Realizado por el Autor.

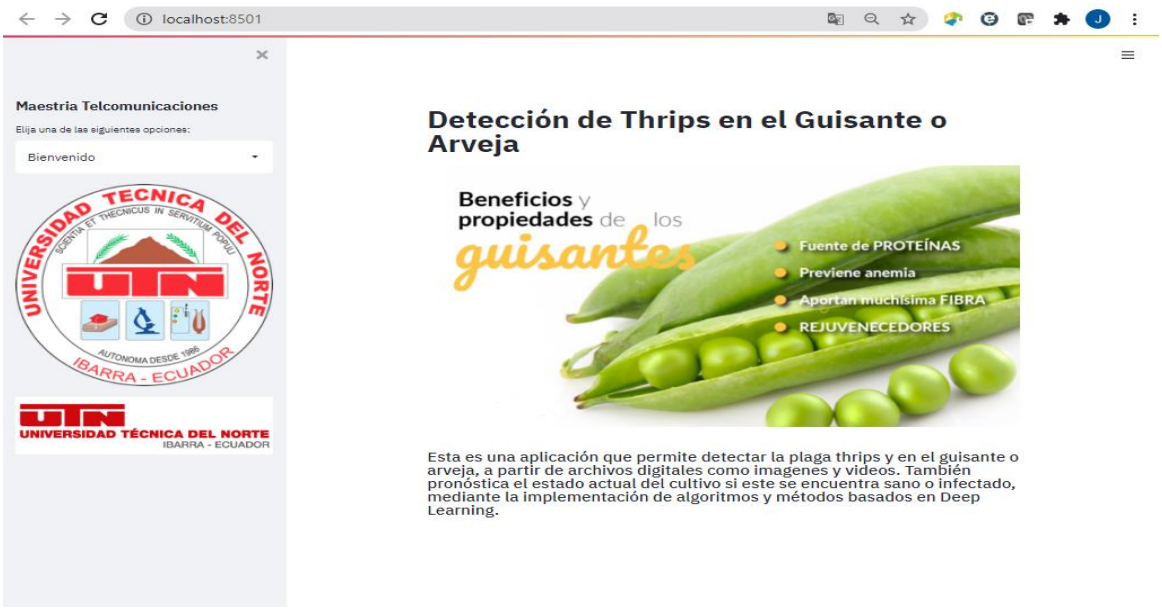
5. Desarrollo de la aplicación del usuario, en este paso se trabajó desde el archivo appDetectionThrips, que mediante la implementación de sentencias de código Python y la librería Streamlit. Permitieron crear diferentes frontales y elementos (como selectbox, butons, imágenes, entre otros), que ayudaron al estilo de cada de las interfaces de la App Web. En el anexo E se encuentra detallado el código utilizado para cada una de las siguientes interfaces:

    - Detección de la plaga thrips en el cultivo del guisante o arveja en imágenes.
    - Detección de la plaga thrips en el cultivo del guisante o arveja videos.
    - Pronostico del estado actual del cultivo del guisante o arveja.

En la Figura 43 se muestra el desarrollo de la App Web, que se utilizará en la detección temprana de la plaga thrips en los cultivos del guisante o arveja.

Figura 43

Entorno de la App Web para la detección de la plaga thrips



Nota: Interfaz de la App Web desarrollada para la detección del objeto de estudio. Realizado por el Autor.

Para ingresar a la aplicación web se deben ejecutar los siguientes pasos:

1. Ingresar el localhost para direccionar a la página web.
2. Seleccionar una de las tareas que se despliegan en el menú o en el selectbox.
3. Dentro de la tarea a ejecutarse se procede a insertar el archivo que será analizado mediante el uso del botón cargar de archivo.
4. Una vez que se haya cargado el archivo, se efectúa la detección automática de la plaga en los cultivos al presionar el botón detección de thrips.

Recursos

Institucionales

Dentro de los recursos institucionales tenemos las siguientes instituciones:

- Universidad Técnica del Norte.
- Bibliotecas Virtuales de Instituciones Académicas.

Humanos

Dentro de los recursos humanos tenemos los siguientes:

- Director de la investigación.
- Asesor de la investigación.
- Investigador.
- Productores del cultivo.

Materiales

Dentro de los recursos humanos tenemos los siguientes:

- Computador portátil.
- Memoria USB.
- Cámara Fotográfica.
- Smartphone.
- Drone.
- SBC (Single Board Computer).
- Resma de papel.

- Materiales de oficina.

Económica (Presupuesto)

En la Tabla 7 se detallan los recursos y materiales necesarios además de los gastos proyectados para el desarrollo de la presente investigación.

Tabla 7

Presupuesto general para el proyecto de investigación

RECURSO	ASPECTOS	UNIDAD	CANTIDAD	COSTO UNITARIO ($)	COSTO TOTAL ($)
Humano	Desarrollo de investigación	Horas/Hombre	150	5,00	750,00
	Recolección de información	Horas/Hombre	20	5,00	100,00
	Desarrollo del sistema	Horas/Hombre	80	5,00	400,00
	Entrevistas con productores	Horas/Hombre	20	15,00	300,00
	Subtotal				**1550,00**
Materiales	Computador portátil	Unidad	1	500	500,00
	Cámara fotográfica	Unidad	1	150	150,00
	Smartphone	Unidad	1	250	250,00
	Drone	Unidad	1	290	290,00
	SBC (Single Board Computer)	Unidad	1	120	120,00
	Materiales de oficina	Unidad	1	120	120,00
	Subtotal				**1430,00**
Otros	Transporte	Pas. /Persona	1	30,00	30,00
	Viáticos	Persona	1	50,00	50,00
	Internet	Unidad	1	25,00	25,00
	Otros	Unidad	1	50,00	50,00
	Subtotal				**155,00**
	Subtotales				3135,00
	Imprevistos				313,00
	TOTAL				**3448,00**

Nota: En la tabla se presenta el presupuesto total de la investigación. Realizado por el Autor.

CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

Funcionamiento del Sistema Desarrollado

Las pruebas de funcionamiento del sistema desarrollado fueron realizadas, sobre una SBC (Raspberry-Pi) la que se constituye de un procesador de cuatro núcleos de 1.5 GHz, una memoria 4GB LPDDR4 en RAM e interfaces de conectividad Wi-Fi con estándar 802.11ac, Bluetooth 5.0 y Gigabit Ethernet.

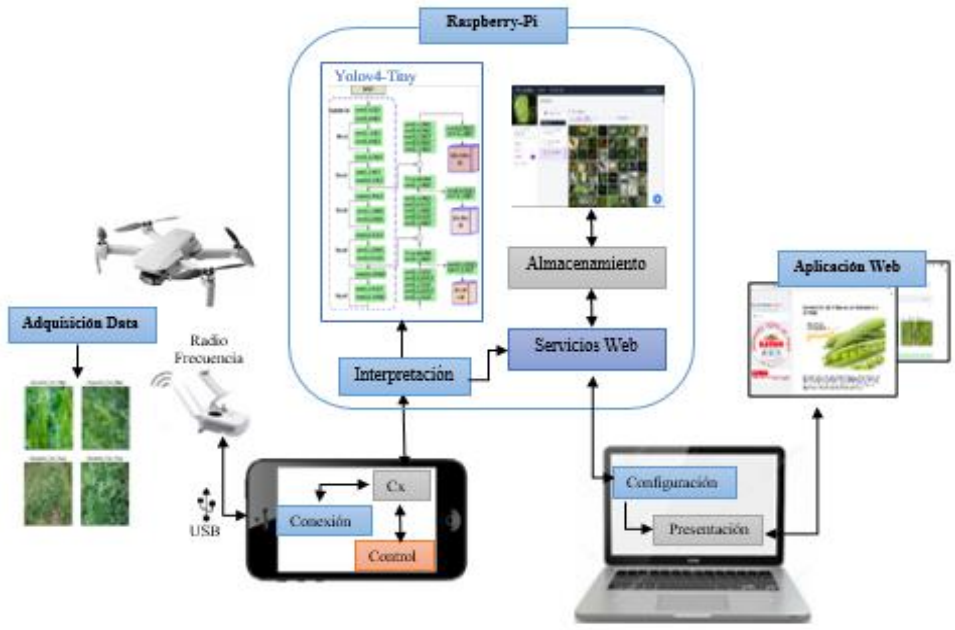
En el entrenamiento de la red yolov4-tiny, se definieron 29 capas convolucionales de predicción en cada una de las imágenes. Donde alcanzó un tiempo promedio de 2 segundos, en la detección del objeto de estudio. El periodo de procesamiento del sistema desarrollado puede ser variable. Debido a que, las imágenes empleadas presentan circunstancias que pueden alterar el resultado, por causas de la cantidad de información y la dimensión que tiene cada una de las imágenes.

Para la detección temprana de la plaga thrips en el guisante o arveja, el sistema realiza un escaneo rápido por toda la imagen. Donde se efectúa la segmentación del objeto de estudio, a través del etiquetado y extracción de la característica, las que permitan el reconocimiento de lesiones necróticas que fueron provocadas por la plaga thrips en los cultivos, mediante la red neuronal yolov4-tiny. Y así determinar la presencia de la misma en los cultivos, a través de ejecución de eventos o tareas desde la App Web.

En la Figura 44 se muestra el diagrama del sistema que será utilizado para para el reconocimiento de lesiones necróticas en los cultivos de la arveja para la detección temprana de la plaga thrips.

Figura 44

Diagrama del sistema de detección de la plaga thrips



Nota: Diagrama del sistema de detección temprana de la plaga thrips en el guisante o arveja. Realizado por el Autor.

Diagrama de Flujo y Control del Sistema Desarrollado

El reconocimiento de lesiones necróticas, para la detección temprana de la plaga thrips, que realiza el sistema desarrollado requiere del intercambio de información ordenada y precisa entre los equipos y dispositivos. De manera que, la comunicación sea correcta y efectiva.

Inicialmente, se ejecuta el encendido del control remoto del drone y la conexión del mismo al smartphone, a través de una interfaz USB. A continuación, se realiza el registro de la ubicación actual del drone, mediante la aplicación DJI Pilot (drone) para proyectar la información al smartphone. Para esta tarea se requiere que el drone haya establecido su ubicación, de manera que se puedan configurar los parámetros básicos de la acción que ejecutará el drone.

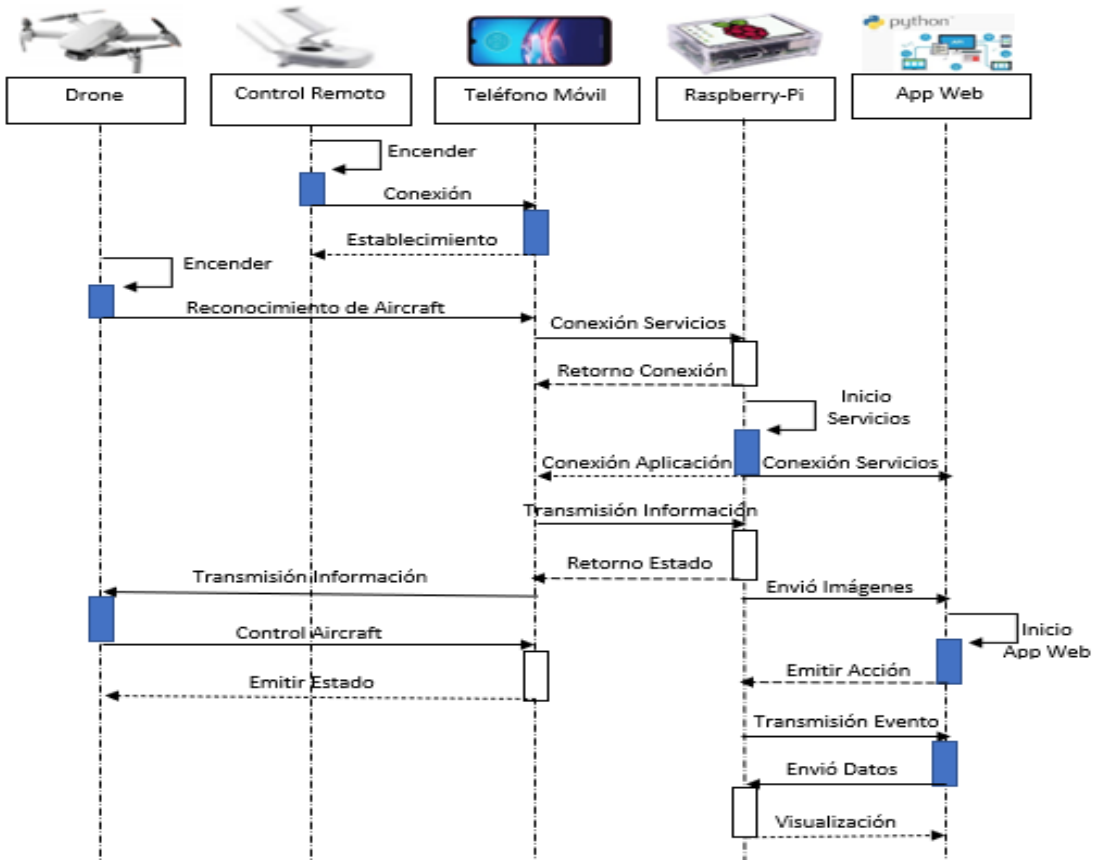
Después que la DJI Pilot, establece comunicación con el drone, se efectúa conexión hacia el Raspberry-Pi (servicios y aplicaciones) este envía una respuesta satisfactoria si la conexión fue exitosa. La App Web del sistema tiene la capacidad de recibir y enviar información por los canales de comunicación de la Raspberry-Pi.

Una vez que la App Web, realizó comunicación esta recibirá las imágenes del guisante o arveja en tiempo real, y ejecutará la tarea de la detección de la plaga thrips en los cultivos. Después de efectuar cada uno de los eventos desde la App Web, estos son enviados hacia la Raspberry-Pi quien se encarga de controlar la comunicación entre el drone y la App Web.

En la Figura 45 se muestra el diagrama de flujo y control de sistema desarrollado para el reconocimiento de lesiones necróticas en los cultivos del guisante o arveja para la detección temprana de la plaga thrips.

Figura 45

Diagrama de flujo y control del sistema desarrollado



Nota: Diagrama de flujo y control del sistema desarrollado para el reconocimiento de lesiones necróticas en el guisante o arveja para la detección temprana de la plaga thrips. Realizado por el Autor.

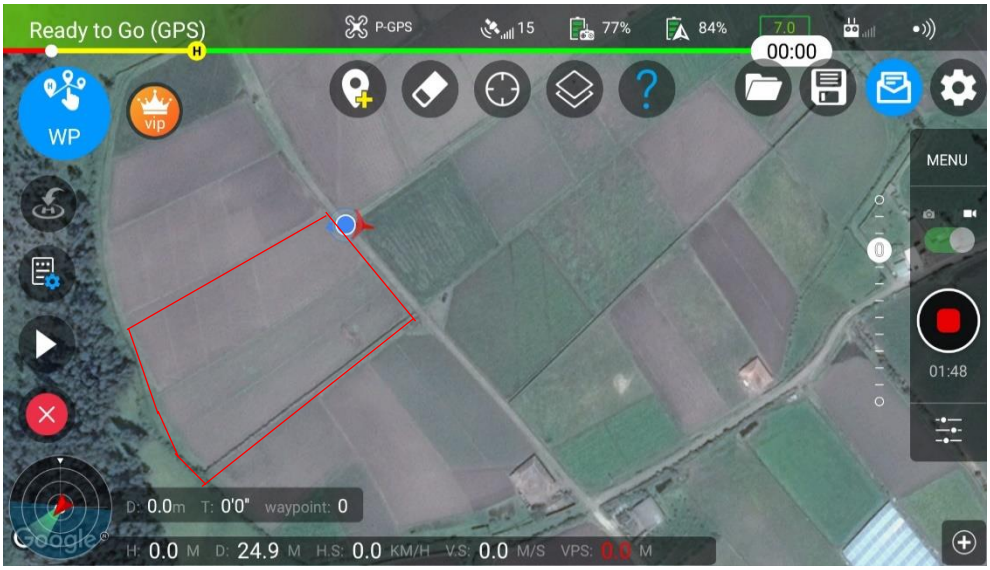
Transmisión de Información del Sistema Desarrollado

Para la transmisión de información se utilizó un drone, el mismo cuenta con un GPS (Sistema de Posicionamiento Global), que provee coordenadas geográficas de latitud y longitud. Esta información será visualizada en el smartphone, a través de la aplicación DJI Pilot, y los servidores de aplicación de mapas como Google Maps, OpenStreetMapm, entre otros. En este caso se decidió utilizar Google Maps, debido a que provee imágenes satelitales y aéreas de alta resolución con un acercamiento de hasta 23 niveles.

Una vez efectuada la geolocalización, como se muestra en la Figura 46 se requiere de la configuración de parámetros como formato de imágenes, altura y velocidad de vuelo. Además, de la tarea o acción que el drone realizará cuando recorra todos los puntos de vuelo, y así iniciar la captura y transmisión de la información sobre la detección del objeto de estudio.

Figura 46

Ubicación de actual del drone en el sitio del objeto de estudio



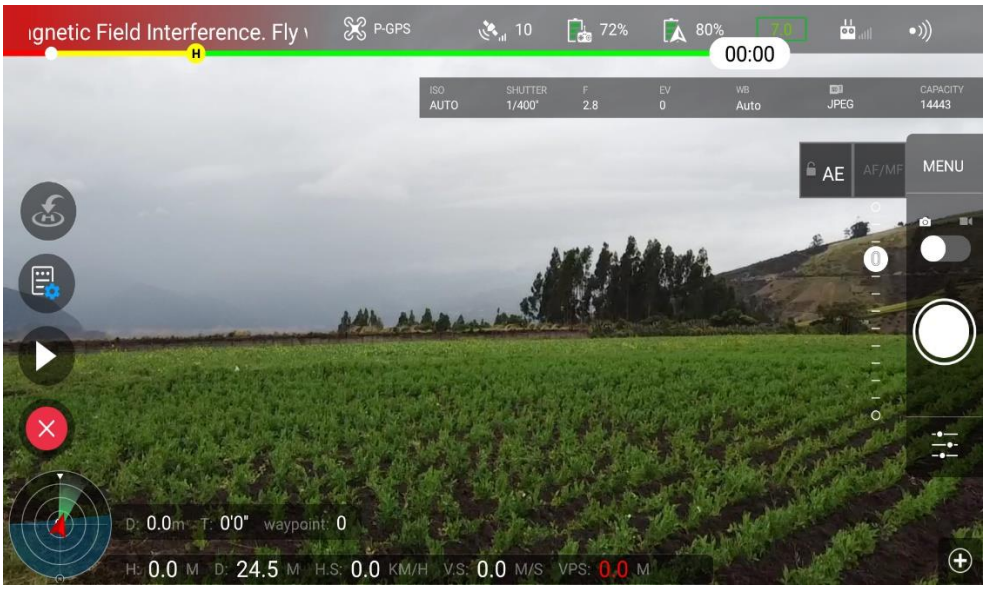
Nota: Ubicación actual donde se realizaron las pruebas de funcionamiento del sistema desarrollado para la detección del objeto de estudio. Realizado por el Autor.

A continuación, el drone inicializa la cámara, la misma que efectúa la captura de imágenes sobre el cultivo, y a través de una conexión de radio frecuencia transmite la información al control remoto. Por medio de una interfaz USB, se habilitará el módulo de codificación el que recibirá las imágenes en formato JPG. En seguida, la imagen es transcodifica para ser visualizada en el smartphone como se muestra en la Figura 47, para

ser inmediatamente enviada al módulo de transmisión (DJI Pilot) para la transferencia de imágenes capturadas en tiempo real al módulo de recepción.

Figura 47

Visualización de las imágenes con la aplicación DJI Pilot



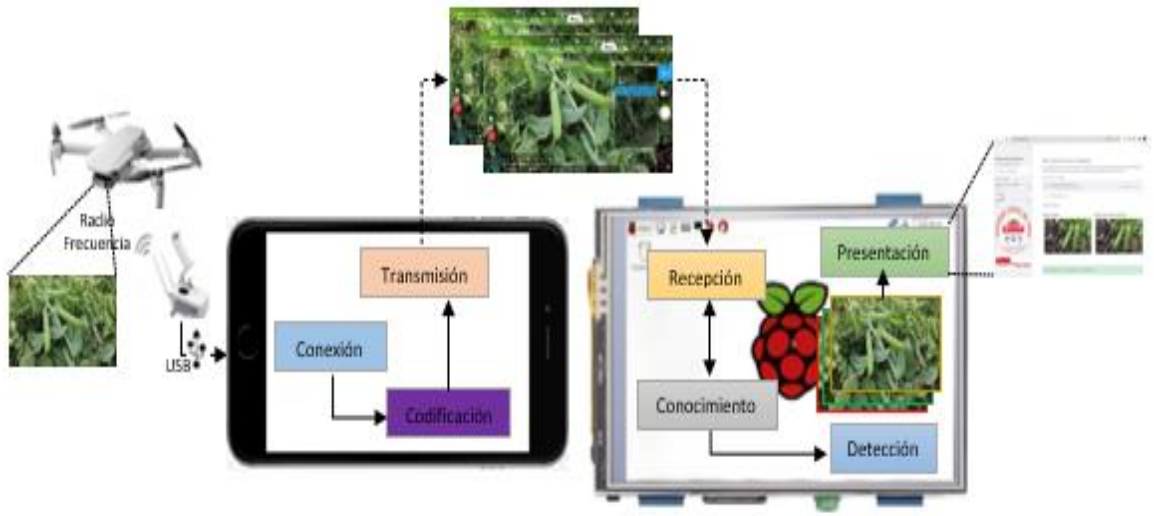
Nota: En la figura se muestra la visualización de las imágenes capturadas con la aplicación DJI Pilot desde el teléfono móvil. Realizado por el Autor.

El reconocimiento de las lesiones necróticas en el cultivo, está dado por la extracción de las características significativas y el etiquetado de las clases, que están en el módulo de conocimiento (*dataset*). A continuación, la información es enviada al módulo de detección (yolov4-tiny), esté analiza la presencia del objeto de estudio en las imágenes y crea como salida un mapa de bits. De manera que, si una imagen que pasa por el módulo de detección tiene lesiones necróticas este graficará un recuadro alrededor de cada lesión necrótica y notifica la información al módulo de conocimiento.

Finalmente, el módulo de detección transfiere un mapa de bits al módulo presentación (App Web), este se encarga de procesarlo para inmediatamente mostrarlo sobre una interfaz web, en la que se mostrará la detección del objeto de estudio. En la Figura 48 se muestra la transmisión de información sobre detección del objeto de estudio.

Figura 48

Transmisión de información en la detección del objeto de estudio



Nota: Integración de los dispositivos para la transmisión de la información sobre la detección del objeto de estudio. Realizado por el Autor.

Análisis de las Pruebas de Funcionamiento

A continuación, se detallan las pruebas de funcionamiento ejecutadas por el sistema de reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips en los cultivos del guisante o arveja:

Caso I: Guisante sin Lesión Necrótica

En la Figura 49 muestra la imagen DSC00048, que fue capturada en las parcelas agrícolas del guisante o arveja. Donde se observó que no existe evidencia alguna sobre la presencia de la plaga thrips de acuerdo al diagnóstico de observación directa que efectuó el experto en la producción del cultivo.

Figura 49

Imagen DSC00048 del guisante capturada desde la parcela

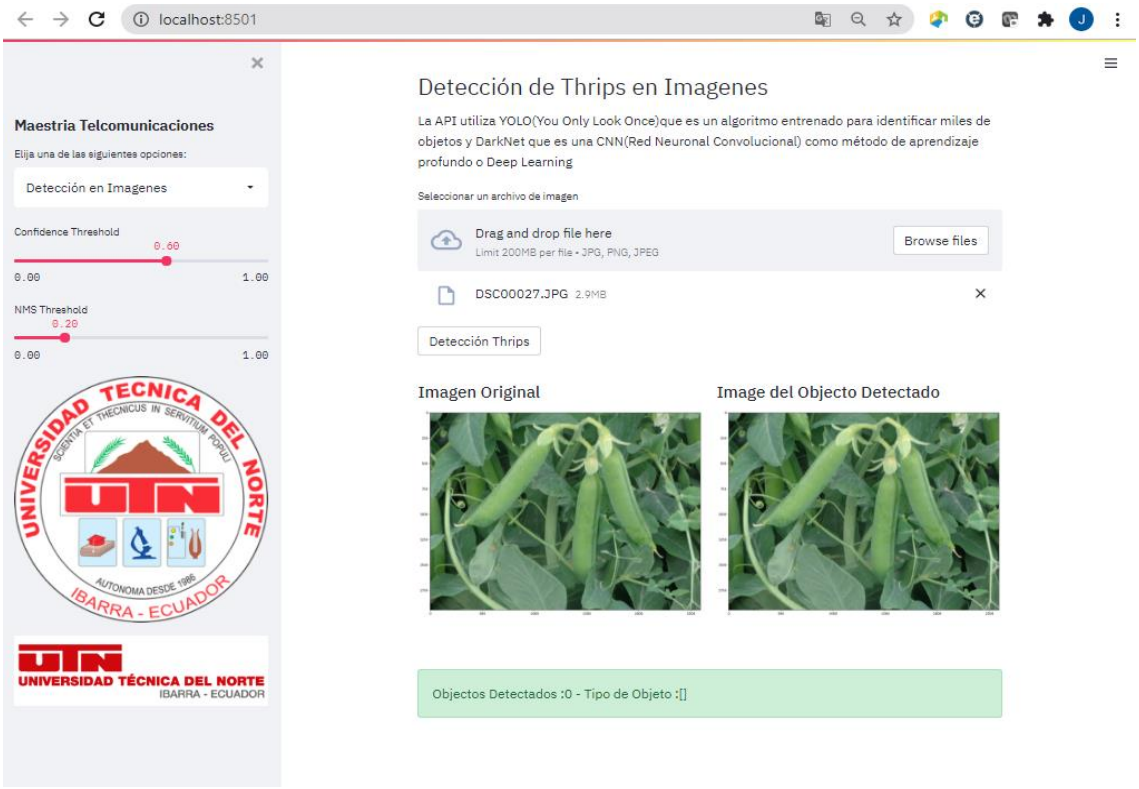
Nota: Imagen DSC00048 del guisante o arveja sin lesiones necróticas. Realizado por el Autor.

Una vez que el sistema analizó la imagen DSC00027, se pudo ver en los resultados que no hay presencia alguna de la plaga thrips en el cultivo como se muestra en la Figura 50. Debido a que, ningún fruto del cultivo presenta lesiones o secuelas ocasionadas por la plaga thrips. Además, existe una alta probabilidad de que el guisante o arveja se encuentre en buen estado y sin infección alguna.

De esta manera, se comprobó que los resultados de la detección temprana de la plaga thrips, realizada por el sistema desarrollado concuerdan con el diagnóstico prescrito del experto. En razón de la observación directa, efectuada a la imagen DSC00027 del cultivo.

Figura 50

Detección de la plaga thrips en la imagen DSC00027

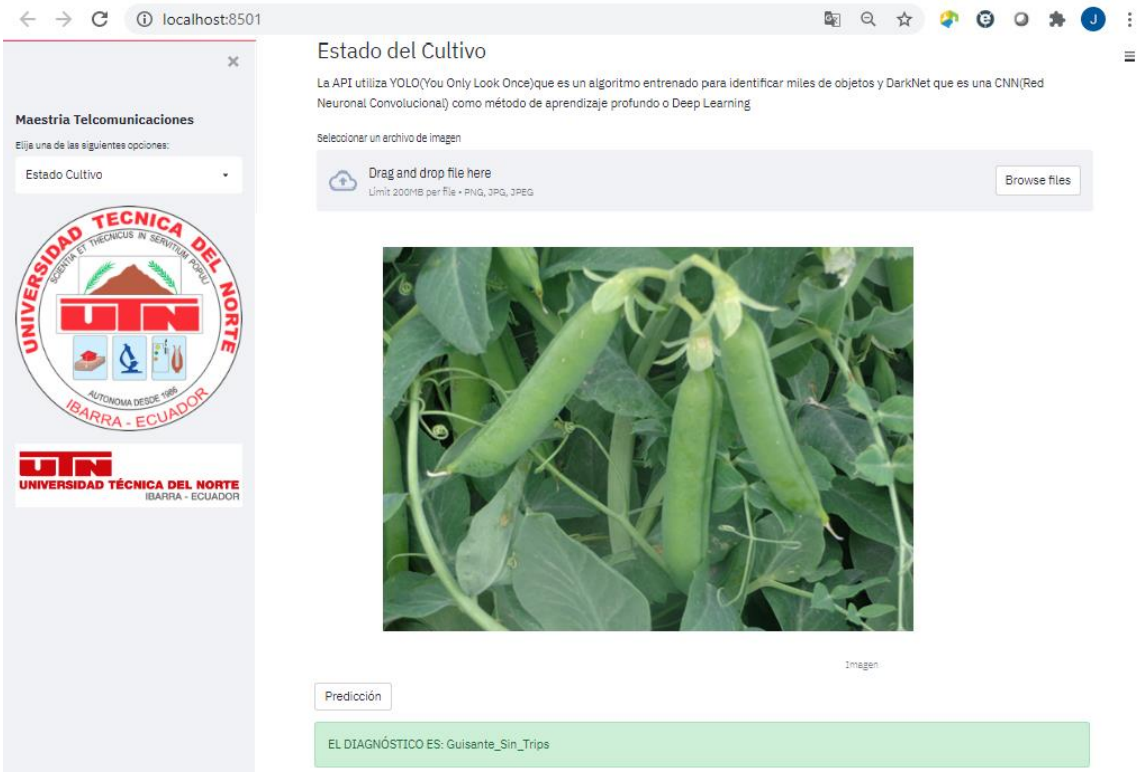


Nota: Detección de lesiones necróticas en el guisante o arveja en la imagen DSC00027. Realizado por el Autor.

Finalmente, en el Caso I, el sistema desarrollado muestra estado actual del cultivo después del análisis realizado a la imagen DSC00027. Donde remite un pronóstico en tiempo real “Guisante_Sin_Thrips”, corroborando así que el cultivo se encuentra en buen estado como se observa en la Figura 51.

Figura 51

Pronóstico del estado actual del cultivo en la imagen DSC00048



Nota: Pronóstico del estado actual del guisante o arveja en la imagen DSC00048. Realizado por el Autor.

Diagnóstico del Experto: en la imagen DSC00027 se observó que no existen lesiones necróticas en el guisante o arveja. Pero se recomienda monitorear el cultivo de forma continua a partir de la fecha que se realizó la detección de la plaga thrips con el sistema desarrollado. Hasta que el cultivo sea cosechado en vista de que en este periodo el guisante o arveja es susceptible a la presencia de la plaga thrips y al ataque de la misma en cualquier momento.

Conclusión del Caso I: una vez analizada la imagen DSC00027 del cultivo, se comprobó que el diagnóstico realizado por el experto en la producción del cultivo concuerda con los resultados finales obtenidos en el sistema desarrollado. Donde se

observó que no hay presencia de la plaga thrips y que todos los frutos del guisante o arveja se encuentran en buen estado.

Caso II: Guisante con Lesión Necrótica

En la Figura 52 muestra la imagen DSC00060, que fue capturada desde las parcelas agrícolas del guisante o arveja. Donde se observó la presencia de lesiones necróticas que fueron causadas por la plaga thrips, de acuerdo al diagnóstico de observación directa que realizó el experto en la producción del cultivo.

Figura 52

Imagen DSC00060 del guisante capturada desde la parcela

*Nota***:** Imagen DSC00060 del guisante o arveja con lesiones necróticas. Realizado por el Autor.

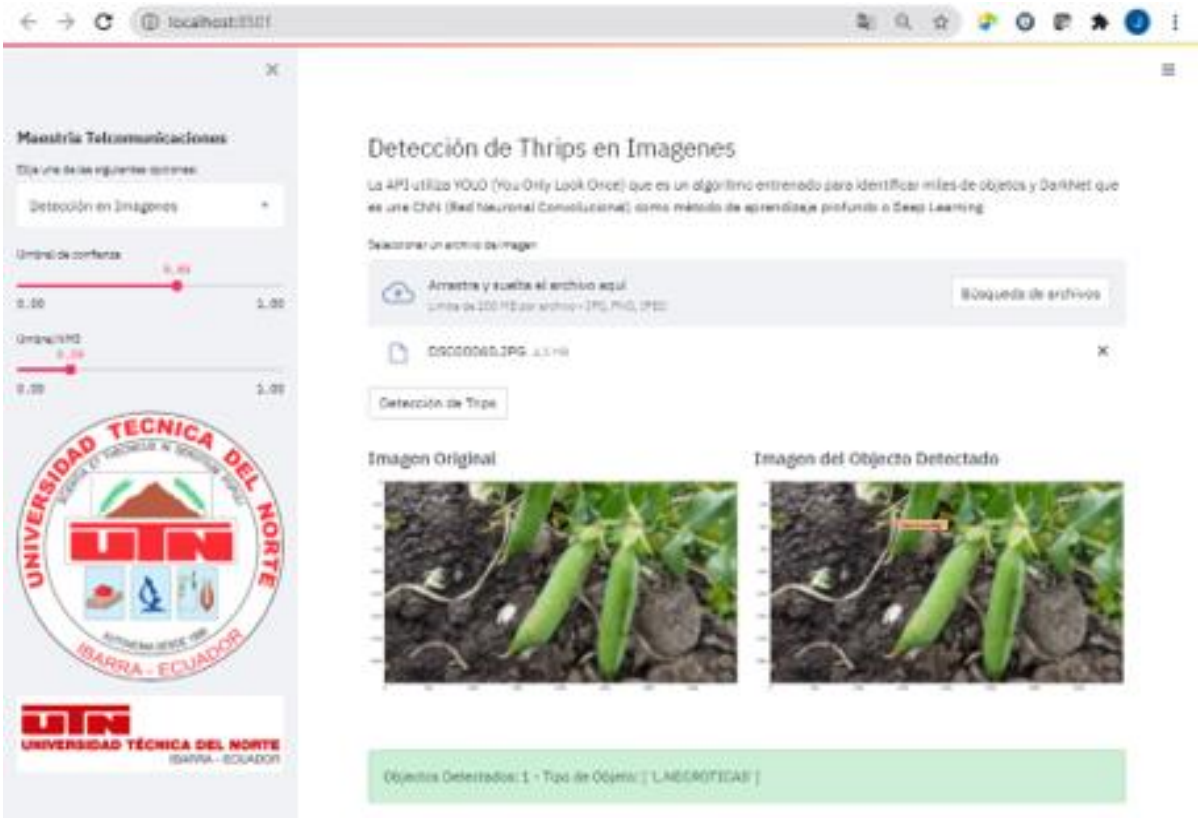
Después que el sistema analizó la imagen DSC00060 se pudo ver en los resultados que el cultivo presenta lesiones necróticas, las mismas que fueron provocadas por la plaga thrips. Este es un factor agravante porque puede afectar al desarrollo del cultivo o causar

la pérdida total del mismo. Además, existe una alta probabilidad que el guisante o arveja se encuentre altamente infectado.

En la Figura 53 se indica el resultado que entrega el sistema desarrollado. El que muestra la presencia de una lesión necrótica en el guisante o arveja, con una precisión del 0.85% en la detección del objeto de estudio.

Figura 53

Detección de la plaga thrips en la imagen DSC00060



Nota: Detección de lesiones necróticas en el guiante o arveja en la imagen DSC00060. Realizado por el Autor.

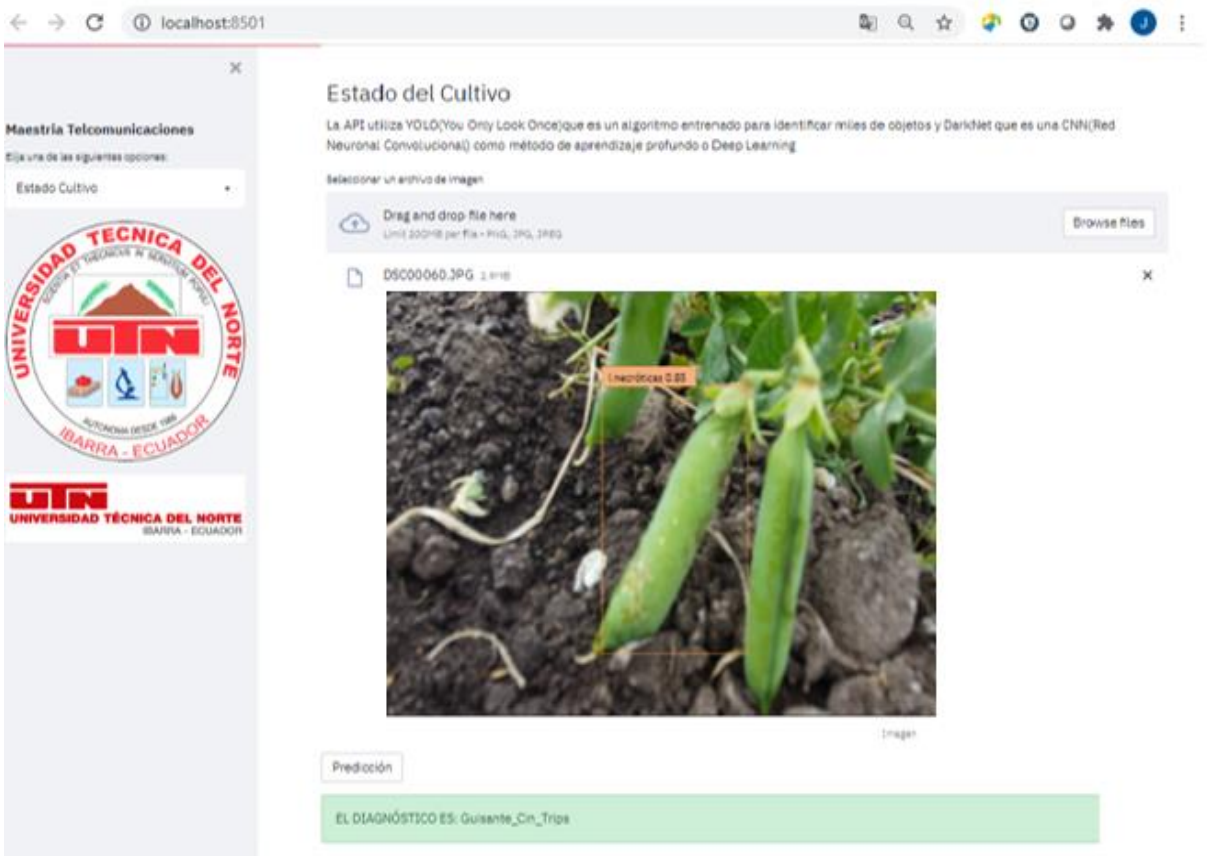
De esta manera, se comprobó que los resultados que se obtuvieron con el sistema desarrollado para la detección temprana de la plaga thrips, concuerdan con el diagnóstico

prescrito del experto. Respecto a las observaciones directas efectuadas, en la imagen DSC00060 del guisante o arveja.

Finalmente, en el Caso II, se muestra el estado actual del cultivo, después que el sistema desarrollado haya realizado el análisis de la imagen DSC00060. Donde remite un pronóstico en tiempo real “Guisante_Con_Thrips”, evidenciando que el cultivo se encuentra infectado con la plaga thrips. En la Figura 54 se observa el resultado de la predicción sobre el estado actual del cultivo.

Figura 54

Pronóstico del estado actual del cultivo en la imagen DSC00060



Nota: Pronóstico del estado actual del guisante o arveja en la imagen DSC00060. Realizado por el Autor.

Diagnóstico del Experto: en la imagen DSC00060 se observa, que existen lesiones necróticas en el guisante o arveja las mismas que fueron provocadas por la plaga thrips. Donde se puede ver que gran parte del fruto se encuentra infectado, debido a este suceso, se recomienda establecer un plan de mitigación de forma inmediata para reducir la propagación de plaga en otras plantas y así evitar la pérdida total del cultivo.

Conclusión del Caso II: una vez analizada la imagen DSC00060, del cultivo se comprobó que el diagnóstico realizado por el experto concuerda con los resultados obtenidos en el sistema desarrollado. Donde se observó la presencia de lesiones necróticas provocadas por la plaga thrips en los frutos del guisante o arveja, ratificando así la existencia de la plaga thrips en los cultivos.

Una vez que ejecutaron las respectivas pruebas de funcionamiento del sistema desarrollado. Se procedió a efectuar un análisis comparativo de los resultados alcanzados por el sistema desarrollado en contraposición al diagnóstico de observación directa al cultivo prescrito por el experto. Y a través, un estudio estadístico de los resultados, se determinará la efectividad y el grado de precisión que tendrá el sistema para la detección del objeto de estudio.

Análisis Comparativo de Resultados

Para el análisis comparativo se generó una tabla de datos, entre cada uno de los resultados alcanzados en las pruebas de funcionamiento realizadas por el sistema desarrollado y los datos del diagnóstico prescrito por el experto en la producción del cultivo. Con el propósito de instituir un margen de error y establecer la confiabilidad del

sistema desarrollado, mediante la consideración de algunos parámetros específicos que permitan tener un sistema más efectivo. Donde la calidad de las imágenes es un parámetro fundamental que influye de forma directa en los resultados del sistema y en el diagnóstico del experto.

Además, para esta tarea se tuvo la colaboración del personal experto en la producción del cultivo. Gracias a que, sus conocimientos y argumentos fueron de gran importancia al momento de diagnosticar el estado actual del guisante o arveja.

En el análisis comparativo de los resultados se eligieron 100 imágenes entre cultivos sanos, infectados o con secuelas de la plaga thrips, los mismos presentaron diferentes valores en los resultados de la detección temprana de la plaga thrips. Las imágenes selectas fueron previamente examinadas por el experto, antes de ser analizadas por el sistema desarrollado. En la Tabla 8 se indican los resultados de detección temprana del objeto de estudio, entregados por el sistema en contraposición a los indicados por el experto.

Tabla 8

Resultados de la detección por el sistema vs el criterio del experto

IMAGEN	**DETECCIÓN DEL SISTEMA**			**CRITERIO DEL EXPERTO**		
	Lesiones	Porcentaje	Estado	Lesiones	Porcentaje	Estado
DSC00001	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00002	1	0,8	Sin_Thrips	0	0,9	Sin_Thrips
DSC00003	2	0,9	Sin_Thrips	0	0,9	Sin_Thrips
DSC00004	0	1	Sin_Thrips	1	0,9	Sin_Thrips
DSC00005	1	0,92	Sin_Thrips	0	1	Sin_Thrips
DSC00006	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00007	0	1	Sin_Thrips	1	1	Sin_Thrips
DSC00008	1	0,91	Sin_Thrips	0	0,9	Sin_Thrips
DSC00009	0	1	Sin_Thrips	0	0,9	Sin_Thrips
DSC00010	1	0,83	Sin_Thrips	0	1	Sin_Thrips
DSC00011	1	0,97	Sin_Thrips	0	1	Sin_Thrips

DSC00012	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00013	1	0,97	Sin_Thrips	1	0,9	Sin_Thrips
DSC00014	1	0,78	Sin_Thrips	0	1	Sin_Thrips
DSC00015	2	0,9	Sin_Thrips	0	1	Sin_Thrips
DSC00016	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00017	1	0,84	Sin_Thrips	1	0,9	Sin_Thrips
DSC00018	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00019	2	0,99	Sin_Thrips	2	0,9	Sin_Thrips
DSC00020	0	1	Sin_Thrips	3	1	Sin_Thrips
DSC00021	1	0,8	Sin_Thrips	1	1	Sin_Thrips
DSC00022	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00023	1	1	Sin_Thrips	0	1	Sin_Thrips
DSC00024	2	0,97	Sin_Thrips	1	1	Sin_Thrips
DSC00025	1	0,92	Sin_Thrips	2	1	Sin_Thrips
DSC00026	0	1	Sin_Thrips	1	1	Sin_Thrips
DSC00027	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00028	2	0,75	Sin_Thrips	0	0,8	Sin_Thrips
DSC00029	1	0,77	Sin_Thrips	1	0,8	Sin_Thrips
DSC00030	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00031	1	0,98	Sin_Thrips	0	1	Sin_Thrips
DSC00032	1	0,97	Sin_Thrips	1	0,9	Sin_Thrips
DSC00033	2	0,75	Sin_Thrips	2	0,8	Sin_Thrips
DSC00034	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00035	1	0,98	Sin_Thrips	2	1	Sin_Thrips
DSC00036	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00037	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00038	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00039	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00040	1	1	Sin_Thrips	1	1	Sin_Thrips
DSC00041	1	0,75	Sin_Thrips	0	0,8	Sin_Thrips
DSC00042	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00043	1	0,75	Sin_Thrips	0	0,8	Sin_Thrips
DSC00044	0	1	Sin_Thrips	1	1	Sin_Thrips
DSC00045	1	0,92	Sin_Thrips	2	1	Sin_Thrips
DSC00046	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00047	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00048	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00049	2	1	Sin_Thrips	3	1	Sin_Thrips
DSC00050	0	1	Sin_Thrips	0	1	Sin_Thrips
DSC00051	2	0,89	Con_Thrips	2	0,9	Con_Thrips
DSC00052	2	0,86	Con_Thrips	2	0,9	Con_Thrips
DSC00053	2	1	Con_Thrips	3	1	Con_Thrips
DSC00054	4	0,97	Con_Thrips	6	1	Con_Thrips
DSC00055	4	0,97	Con_Thrips	6	1	Con_Thrips
DSC00056	3	0,98	Con_Thrips	3	1	Con_Thrips
DSC00057	4	0,93	Con_Thrips	3	1	Con_Thrips
DSC00058	3	0,97	Con_Thrips	5	1	Con_Thrips
DSC00059	2	0,9	Con_Thrips	3	1	Con_Thrips
DSC00060	2	0,60	Con_Thrips	2	0,9	Con_Thrips
DSC00061	2	0,75	Con_Thrips	2	0,8	Con_Thrips
DSC00062	1	1	Con_Thrips	1	1	Con_Thrips
DSC00063	2	0,88	Con_Thrips	3	0,9	Con_Thrips
DSC00064	1	0,89	Con_Thrips	3	0,9	Con_Thrips
DSC00065	2	0,91	Con_Thrips	3	1	Con_Thrips
DSC00066	1	0,99	Con_Thrips	2	1	Con_Thrips
DSC00067	2	0,84	Con_Thrips	3	0,9	Con_Thrips
DSC00068	3	0,97	Con_Thrips	4	1	Con_Thrips
DSC00069	3	0,95	Con_Thrips	4	1	Con_Thrips
DSC00070	2	0,94	Con_Thrips	4	1	Con_Thrips

DSC00071	1	0,99	Con_Thrips	1	1	Con_Thrips
DSC00072	1	0,99	Con_Thrips	2	1	Con_Thrips
DSC00073	1	0,94	Con_Thrips	1	1	Con_Thrips
DSC00074	2	0,75	Con_Thrips	2	0,8	Con_Thrips
DSC00075	1	0,97	Con_Thrips	2	1	Con_Thrips
DSC00076	1	0,89	Con_Thrips	1	0,9	Con_Thrips
DSC00077	2	0,81	Con_Thrips	2	0,9	Con_Thrips
DSC00078	4	0,98	Con_Thrips	5	1	Con_Thrips
DSC00079	1	0,91	Con_Thrips	1	1	Con_Thrips
DSC00080	2	0,99	Con_Thrips	2	1	Con_Thrips
DSC00081	3	0,98	Con_Thrips	5	1	Con_Thrips
DSC00082	1	0,92	Con_Thrips	2	1	Con_Thrips
DSC00083	2	0,81	Con_Thrips	2	1	Con_Thrips
DSC00084	1	0,96	Con_Thrips	2	1	Con_Thrips
DSC00085	2	0,98	Con_Thrips	3	1	Con_Thrips
DSC00086	1	0,9	Con_Thrips	2	1	Con_Thrips
DSC00087	1	0,99	Con_Thrips	2	1	Con_Thrips
DSC00088	1	0,97	Con_Thrips	2	1	Con_Thrips
DSC00089	2	0,88	Con_Thrips	1	0,9	Con_Thrips
DSC00090	2	0,99	Con_Thrips	2	1	Con_Thrips
DSC00091	1	1	Con_Thrips	1	1	Con_Thrips
DSC00092	1	1	Con_Thrips	2	1	Con_Thrips
DSC00093	1	1	Con_Thrips	2	1	Con_Thrips
DSC00094	1	0,99	Con_Thrips	3	1	Con_Thrips
DSC00095	2	0,85	Con_Thrips	3	0,9	Con_Thrips
DSC00096	3	0,79	Con_Thrips	5	0,9	Con_Thrips
DSC00097	4	0,98	Con_Thrips	6	1	Con_Thrips
DSC00098	2	0,72	Con_Thrips	2	0,8	Con_Thrips
DSC00099	2	0,87	Con_Thrips	3	0,9	Con_Thrips
DSC00100	2	0,99	Con_Thrips	4	1	Con_Thrips

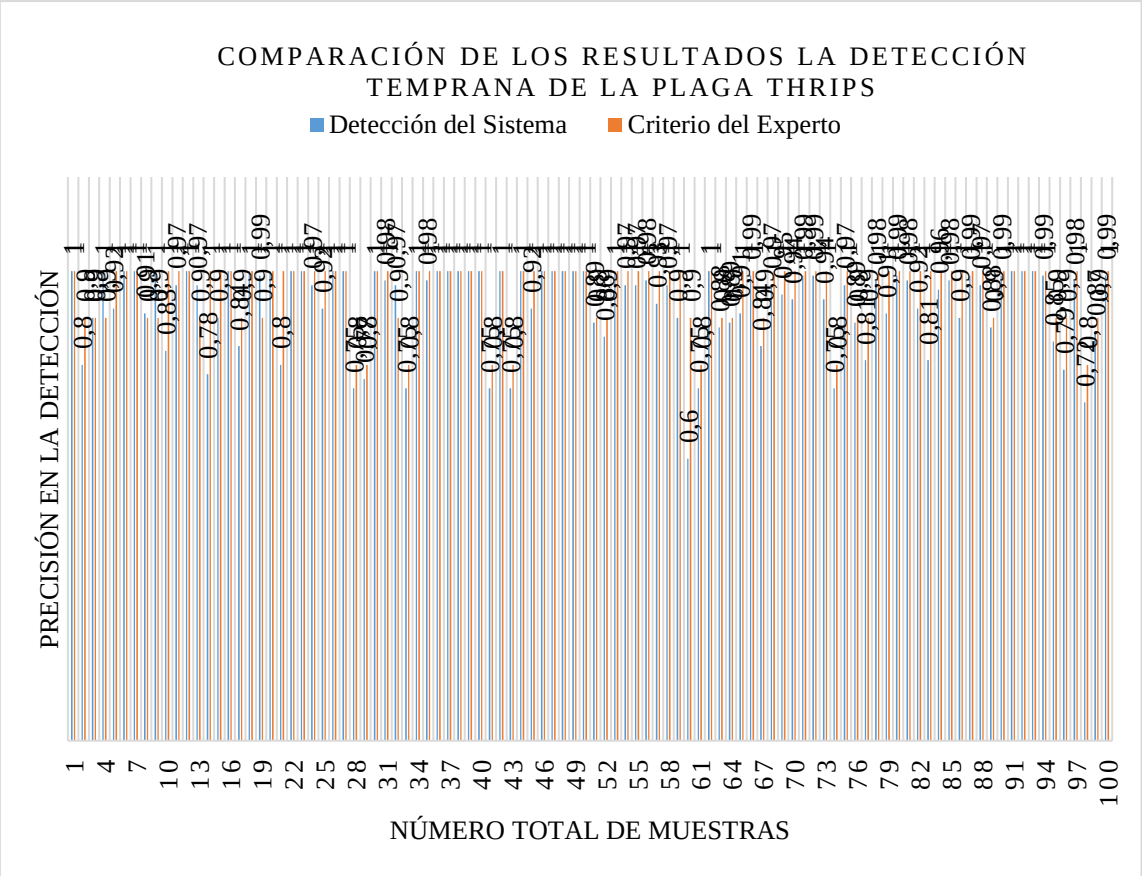
Nota: En la tabla se muestran los valores detectados por el sistema vs los realizados por el experto. Realizado por el Autor.

Después de realizar la comparación de los resultados, de la detección temprana de la plaga thrips en el guisante o arveja. Se observa que existe una ligera variación entre los valores, que entrega el sistema respecto al diagnóstico señalado por el experto. Para este tipo de análisis es muy importante tener en cuenta que el resultado puede variar de acuerdo al criterio de las personas que participen en las observaciones directas a los cultivos.

En la Figura 55 se muestra la comparación de los resultados de la detección temprana de la plaga thrips en el guisante o arveja, que entregó el sistema desarrollado en contraposición al diagnóstico señalado por el experto.

Figura 55

Comparación de los resultados de la detección de la plaga thrips



Nota: Diagrama del análisis comparativo de los resultados de la detección temprana de la plaga thrips en el guisante o arveja. Realizado por el Autor.

Cálculo de la Eficiencia del Sistema Desarrollado

Una vez efectuado el análisis comparativo de los resultados indicados por el experto en la producción del cultivo. Se consideró que, el error humano determinado en la técnica de observación directa al guisante o arveja es del 5%. En cambio, la efectividad del sistema desarrollado depende de factores externos, los mismos que de alguna u otra forma pueden alterar los resultados, mediante la presencia de falsos positivos.

Ecuación 3

Cálculo del error del experto en la detección del objeto

$$Error\ Experto = \frac{(N.\ de\ detecciones) * 5}{100}$$

$$Error\ Experto = \frac{164 * 5}{100}$$

$$Error\ Experto = 8{,}20$$

A continuación, se muestra el cálculo del error que tiene el sistema desarrollado para la detección temprana de la plaga thrips en los cultivos.

Ecuación 4

Cálculo del error del sistema en la detección del objeto

$$Error\ Sistema = \frac{(N.\ de\ detecciones\ no\ realizadas) * 0{,}05}{Error\ del\ Experto}$$

$$Error\ Sistema = \frac{33\ * 0{,}05}{8{,}20}$$

$$Error\ Sistema = 0{,}2012$$

$$\%Error\ Sistema = 20\%$$

Después de realizar el cálculo del error del sistema desarrollado se pudo ver que el resultado es del 20%, respecto a las detecciones no realizadas, donde la efectividad es de 80%. Este resultado permite que el sistema desarrollado sea confiable, sobre todo cuando se utilizan imágenes que cumplan con los parámetros requeridos. Además, que en el diagnóstico prescrito por el experto se excluya los falsos positivos.

Comprobación Estadística de la Hipótesis

Para la comprobación estadística de la hipótesis, se utilizó el modelo chi cuadrado a través del uso de las distribuciones esperadas sobre las detecciones de la plaga thrips en el cultivo, de las realizadas por el sistema desarrollado, así como las observaciones indicadas por experto en la producción del cultivo. Además, se efectuó la prueba de bondad de ajuste del modelo chi cuadrado con la finalidad de evidenciar que el número de muestras ejecutadas se ajusten al valor de la distribución teórica.

El cálculo del modelo estadístico chi cuadrado fue realizado con la aplicación de la siguiente ecuación.

Ecuación 5

Cálculo del modelo estadístico chi cuadrado

$$X_{cal}^2 = \sum \frac{(fo - fe)^2}{fe}$$

Dónde:

$X_{cal}^2 =$ modelo chi cuadrado.

f_o = frecuencia del valor observado.

f_e = frecuencia del valor esperado.

En cambio, para el cálculo de la frecuencia esperada se utilizó la siguiente ecuación.

Ecuación 6

Cálculo de la frecuencia esperada para el modelo chi cuadrado

$$f_e = \frac{\text{Total columna (para dicha celda)} * \text{Total de fila (para dicha celda) Suma Total}}{T_o}$$

Dónde:

T_o = total de observaciones.

En la Tabla 9 se indica el resultado sobre el cálculo de la frecuencia esperada y del modelo chi cuadrado para la detección de la plaga thrips en el guisante o arveja.

Tabla 9

Resultados del cálculo de la frecuencia espera y chi cuadrado

Observación (T_o)	**Frecuencia Observada Sistema (f_o)**	**Frecuencia Observada Experto (f_o)**	**Frecuencia Esperada (f_t)**	**Chi Cuadrado (X^2_{cal})**
1	78	66	77,95813	2,150106203
2	69	59	69,29611	1,911205514
3	54	46	54,13759	1,493129308
4	84	71	83,91326	2,314350427
5	88	74	87,70289	2,418869479
6	98	83	97,98903	2,702564047
7	72	61	72,00299	1,985861979

Nota: En la tabla se indican los valores de la frecuencia esperada para la detección y modelo estadístico chi cuadrado. Realizado por el Autor.

Aplicando la Ecuación 5, se muestra el resultado del modelo estadístico chi cuadrado, en base a las observaciones esperadas y realizadas por el sistema, además de las efectuadas por el experto.

$$X^2_{cal} = \sum \frac{(fo - fe)^2}{fe}$$

$$X_{cal}^2 = 14{,}97608696$$

Una vez realizado el cálculo del modelo estadístico chi cuadrado se consideró evaluar la hipótesis nula para un nivel de significancia de 5%, que equivale a un nivel de confianza de 95%.

Para comprobar si la hipótesis nula, va a ser aceptada o no, se requiere conocer el grado de libertad del modelo estadístico chi cuadrado. El cálculo del grado de libertad se realiza a través de la siguiente ecuación.

Ecuación 7

Cálculo del grado de libertad para el modelo chi cuadrado

$$v = (Total\ de\ filas - 1) * (Total\ de\ columnas - 1)$$

$$v = (7 - 1) * (2 - 1)$$

$$v = (6) * (1)$$

$$v = 6$$

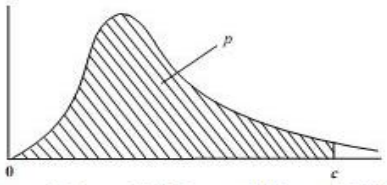
Comprobación del Valor Teórico

A continuación, se procedió a determinar el valor teórico o crítico mediante el uso de la distribución del modelo estadístico chi cuadrado. En la Figura 56 se muestra un fragmento de la distribución chi cuadrado. Donde el valor de la fila (6) representa el grado de libertad, mientras que el valor de la columna (0,95) corresponde el nivel de confianza.

Figura 56

Tabla de la distribución Chi Cuadrado

$p = \mathrm{P}(X \leq c)$



p	0,005	0,01	0,025	0,05	0,1	0,9	0,95	0,975	0,99	0,995
$\nu = 1$	0,00004	0,0002	0,001	0,004	0,016	2,706	3,841	5,024	6,635	7,879
2	0,010	0,020	0,051	0,103	0,211	4,605	5,991	7,378	9,210	10,597
3	0,072	0,115	0,216	0,352	0,584	6,251	7,815	9,348	11,345	12,838
4	0,207	0,297	0,484	0,711	1,064	7,779	9,488	11,143	13,277	14,860
5	0,412	0,554	0,831	1,145	1,610	9,236	11,070	12,833	15,086	16,750
6	0,676	0,872	1,237	1,635	2,204	10,645	12,592	14,449	16,812	18,548
7	0,989	1,239	1,690	2,167	2,833	12,017	14,067	16,013	18,475	20,278
8	1,344	1,646	2,180	2,733	3,490	13,362	15,507	17,535	20,090	21,955
9	1,735	2,088	2,700	3,325	4,168	14,684	16,919	19,023	21,666	23,589
10	2,156	2,558	3,247	3,940	4,865	15,987	18,307	20,483	23,209	25,188
11	2,603	3,053	3,816	4,575	5,578	17,275	19,675	21,920	24,725	26,757
12	3,074	3,571	4,404	5,226	6,304	18,549	21,026	23,337	26,217	28,300

Nota: En la figura se muestra la tabla de valores de la distribución Chi Cuadrado. Tomado desde *TABLA PARA VALORES DE CHI-CUADRADO CRÍTICO* [Imagen], slideplayer, 2016, www. slideplayer.es/slide/10624268/35/images/17/TABLA+PARA+VALORES+DE+CHI-CUADRADO+CR%C3%8DTICO.jpg

En vista de que, el valor teórico o crítico es 12,592 respecto al valor calculado que es 14,97608696; de esta manera se puede aseverar que hay evidencia estadística para rechazar la hipótesis nula.

$$X^2_{cal} \leq X^2_{critico}$$

$$14{,}97608696 \leq 12{,}592$$

Una vez revisado el valor teórico o crítico se menciona que “si el valor calculado en menor al valor teórico, la hipótesis nula es aceptada caso contrario la hipótesis es rechazada”.

Después de la comparación realizada entre el valor calculado y el valor teórico o crítico del modelo chi cuadrado, se comprobó que la condición anterior no se cumple. Por lo tanto, la hipótesis que fue planteada en el presente trabajo de investigación será considerada como verdadera.

CONCLUSIONES

- La implementación de los modelos o técnicas de aprendizaje profundo o deep learning, en el desarrollo de sistemas aplicados a la agricultura de precisión tienen un alto nivel de aceptación. Debido a que son empleados en algunas tareas como el monitoreo del estado del cultivo o la predicción de las cosechas. En este caso permitió efectuar el reconocimiento de lesiones necróticas en el guisante o arveja de manera rápida y efectiva, a la vez pudo notificar a los productores sobre la presencia de la plaga thrips en sus cultivos. Con la finalidad de que puedan ejecutar un plan de mitigación que evite la pérdida total del guisante o arveja en las parcelas agrícolas.
- En el desarrollo del sistema se utilizaron imágenes que fueron capturadas desde las parcelas agrícolas del guisante o arveja, las mismas que se procesaron y analizaron con Roboflow. Esta herramienta permitió efectuar el tratamiento de algunos factores como el ruido digital, que de alguna forma pueden modificar los resultados alcanzados en la detección del objeto de estudio. Gracias al tratamiento de las imágenes con Roboflow se consiguió un dataset eficaz y confiable.
- El entrenamiento de la arquitectura yolov4-tiny, fue una solución que permitió alcanzar rapidez y eficiencia en la detección de la plaga thrips en tiempo real, debido al nivel de la intersección sobre la unión (IoU) que es del 59,23% para la

una precisión media (mAP) que es de 87,8% sobre un conjunto de datos de alta densidad. De esta manera Yolov4, permite contraponerse a otros algoritmos de alta precisión como Faster-RCNN, donde permitió determinar de forma correcta el estado actual del guisante o arveja a partir de la integración de dataset y el entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29.

- El sistema desarrollado ejecuta el reconocimiento de las lesiones necróticas para la detección temprana de la plaga thrips con una efectividad del 85%, según los resultados alcanzados en el caso II. mientras que en el caso I la detección tuvo una efectividad del 100%. Además, logró efectuar la predicción sobre el estado actual del guisante o arveja de forma acertada. Sin embargo, es necesario contar con un diagnóstico final del experto en la producción del cultivo.
- En el análisis comparativo, de cada una de las detecciones realizadas con el sistema desarrollado, en contraposición al diagnóstico prescrito por el experto a las observaciones directas efectuadas al cultivo. Se observó que hay una variación significativa entre los dos resultados, lo que garantiza que el sistema puede ser bastante confiable. Debido a que, el cálculo del error del sistema es del 20% sobre aquellas detecciones no realizadas, determinando así que la efectividad del sistema desarrollado es del 80%.

RECOMENDACIONES

- El presente trabajo de investigación puede ser utilizado en el sector agrícola y sobre todo en estudios similares basados en aprendizaje automático. Para lo que se recomienda el uso de la herramienta Python como entorno de desarrollo principal, debido su amplia gama de librerías que tiene lo que la hacen que se

convierta en una de las principales alternativas para la integración de arquitecturas empleadas en la detección y clasificación del objeto de estudio y para el desarrollo de aplicaciones basadas en el campo del deep learning.

- En la creación del dataset se debe recurrir al criterio del personal experto en el área, debido a que permitirá seleccionar de manera adecuada las imágenes de los cultivos que serán utilizadas tanto para el análisis como en el entrenamiento de la red neuronal convolucional yolov4-tiny. Porque si no cumplen con los parámetros mínimos de calidad, pueden ocasionar el aumento de falsos positivos en la detección del objeto de estudio y provocar el incremento en el margen de error del sistema.
- Para el entrenamiento de la red neuronal yolov4-tiny, se debe tener en cuenta la cantidad de imágenes que se utilizarán en esta tarea, debido a que pueden inferir de manera directa en la detección de la plaga thrips. Por lo que se recomienda realizar la configuración de ciertos parámetros fundamentales que se encuentran en el archivo yolov4.tiny.cfg, el mismo que tiene información referente a la arquitectura yolov4-tiny como el número de épocas, cantidad de imágenes que serán entrenadas en cada época, números de filtros que se utilizarán y la cantidad de clases que se predecirán, de manera que los resultados que se obtengan sean más precisos y apegados a la realidad.
- Es fundamental que para este tipo de sistemas siempre se debe conocer un margen de error, ya que es necesario saber el nivel de confiabilidad que ofrece a los usuarios, sobre todo si va a ser empleado como una herramienta de diagnóstico, puesto que siempre debe ir acompañado del criterio de una persona experta en el área o tema donde se lo esté evaluando o utilizando al sistema.

- Para investigaciones futuras se sugiere escalar el nivel de programación que tiene el sistema, de forma que permita ampliar la detección de otras plagas o enfermedades que atacan al guisante o arveja, y a otros cultivos. De manera que, se aprovechen los beneficios que ofrece el deep learning al sector agrícola.

GLOSARIO

APP Application Program (Programación de Aplicación)

CEPAL Economic Commission for Latin America and the Caribbean (Comisión Económica para América Latina y el Caribe)

CNN Neural Networks Convolutionals (Redes Neuronales Convolucionales)

CSPNet Cross Stage Partial Networks (Redes Parciales entre Etapas)

DL Deep Learning **(**Aprendizaje Profundo)

FAO Food and Agriculture Organization (Organización de las Naciones Unidas para la Alimentación y la Agricultura)

Fast R-CNN Rápidas Redes Neuronales Convolucionales Basadas en Regiones

FPS Frames per Second (Fotogramas por Segundo)

GPU Graphics Processing Unit (Unidad de procesamiento Gráfico)

JPG Joint Photographic Experts Group (Grupo de Expertos Fotográficos Conjuntos)

IA Artificial Intelligence (Inteligencia Artificial)

IICA Instituto Interamericano de Cooperación para la Agricultura

IoT Internet of Things (Internet de las Cosas)

LAC Latin America and the Caribbean (América Latina y el Caribe)

LPDDR4 Low-Power Double Data Rate (Bajo consumo de energía doble velocidad de datos)

ML Machine Learning (Aprendizaje de Maquinas)

MQTT Mesaré Queue Telemetry Transport (Protocolo usado para la comunicación machine-to-machine (M2M) en el Internet of Things)

PANet Path Aggregation Network (Red de Agregación de Rutas)

PND	Plan Nacional de Desarrollo
RAM	Random Access Memory (Memoria de Acceso Aleatorio)
R-CNN	Redes Neuronales Convolucionales Basadas en Regiones
RELU	Rectified Linear Unit (Unidad lineal rectificada)
RoI	Region of Interest (Región de Interés)
RPN	Region Proposal Network (Red de Propuesta de Región)
SBC	Single Board Computer (Computadora de Placa Única)
SDD	Single Short Multibox Detector (Detector de caja múltiple de disparo único)
SPP	Spatial Pyramid Pooling (Agrupación Piramidal Espacial)
VGG	Visual Geometry Group (Grupo Geométrico Visual)
WEB	World Wide Web
YOLO	You Only Look Once (Solo se Mira una Vez)

REFERENCIAS

Agronomija. (s.f.). *GRAŠKOV TRIPS – KAKOTHRIPS ROBUSTUS*. Obtenido de https://agronomija.rs/: https://agronomija.rs/wp-content/uploads/2013/12/Kakothrips-robustus.jpg

Ai, Y., Sun, C., Tie, J., & Cai, X. (21 de Septiembre de 2020). *Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments*. Obtenido de https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/9201298

AnalyticsVidhya. (24 de Febrero de 2020). *YOLOv4 VS YOLOv4-tiny*. Obtenido de https://medium.com: https://medium.com/analytics-vidhya/yolov4-vs-yolov4-tiny-97932b6ec8ec

Angulo Pérez, A. (2019). *Identificación de las principales plagas y enfermedades en el cultivo de arveja (Pisum sativum), parroquia Bolívar, cantón Bolívar, Provincia del Carchi*. Obtenido de www.dspace.utb.edu.ec: http://dspace.utb.edu.ec/bitstream/handle/49000/6395/E-UTB-FACIAG-ING%20AGRON-000152.pdf?sequence=1&isAllowed=y

Bereinsten, R., & Edan, Y. (02 de Marzo de 2017). *Automatic Adjustable Spraying Device for Site-Specific Agricultural Application.* Obtenido de www.ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/7869323

Berrones Reyes, M. C. (Julio de 2019). *Clasificación de mamografías mediante redes neuronales convolucionales*. Obtenido de www.eprints.uanl.mx: http://eprints.uanl.mx/17656/1/1080288627.pdf

Bo Gong, Daji Ergu, Ying Cai, & Bo Ma. (Septiembre de 2020). *A Method for Wheat Head Detection Based on Yolov4*. Obtenido de www.assets.researchsquare.com: https://assets.researchsquare.com/files/rs-86158/v1_stamped.pdf?c=1602118667

Burgués Miró, J. (Junio de 2019). *Recognition of musical symbols in scores using neural networks.* Obtenido de www.upcommons.upc.edu: https://upcommons.upc.edu/bitstream/handle/2117/165583/Final_Report.pdf

Cabezas Gómez, E. V. (2019). *Reconocimiento de patrones de imágenes médicas para establecer diagnósticos previos en trastornos pulmonares.* Obtenido de https://repositorio.uta.edu.ec: https://repositorio.uta.edu.ec/bitstream/123456789/29180/1/Tesis_t1533masc.pdf

CEPAL. (Febrero de 2020). *Perspectivas de la agricultura y del desarrollo rural en las Américas: una mirada hacia América Latina y el Caribe 2019-2020*. Obtenido de www.repositorio.cepal.org: https://repositorio.cepal.org/

CGSpace. (Noviembre de 2018). *Retos del Cambio Climático para la Agricultura en América Latina y el Caribe.* Obtenido de www.cgspace.cgiar.org: https://cgspace.cgiar.org/rest/bitstreams/160740/retrieve

Chen, C. J., Huang, Y. Y., Li, Y. S., Chen, Y. C., & Cha, C. Y. (01 de Febrero de 2021). *Identificación de plagas de árboles frutales con aprendizaje profundo en drones integrados para lograr una fumigación precisa de pesticidas*. Obtenido de https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/9343827

Dalai, R., & Senapati, K. K. (13 de Febrero de 2020). *Un sistema de detección de plagas basado en visión inteligente que utiliza un mecanismo de aprendizaje profundo basado en RCNN*. Obtenido de https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/8995072

Delgado Chamorro, C. G. (2014). *Efecto del ácido acetilsalicílico para activación de defensas en el cultivo de arveja (Pisum sativum),en el sector de Chapués, cantón Tulcán, Carchi – Ecuador*. Obtenido de www.repositorio.upec.edu.ec: http://repositorio.upec.edu.ec/bitstream/123456789/240/1/199%20EFECTO%20DEL%20%C3%81CIDO%20ACETILSALIC%C3%8DLICO%20PARA%20ACTIVACI%C3%93N%20DE%20DEFENSAS%20EN%20EL%20CULTIVO%20DE%20ARVEJA%2

0%28PISUM%20SATIVUM%29%2C%20EN%20EL%20SECTOR%20DE%20CHAPU%C3%89S%2

Do Thuan. (2021). *Evolution of YOLO Algorithm and YOLOv5: The State-of-the-art Object Detection Algorithm*. Obtenido de www.theseus.fi: https://www.theseus.fi/bitstream/handle/10024/452552/Do_Thuan.pdf?isAllowed=y&sequence=2

Elgendy, M. (2020). *Deep Learning for Vision Systems* .

FutureSpace. (17 de Marzo de 2021). *Redes Neuronales y Deep Learning*. Obtenido de www.futurespace.es: https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-1-preludio/

GAD Municipal del Cantón Bolívar. (s.f.). *GAD Municipal del Cantón Bolívar*. Obtenido de www.municipiobolivar.gob.ec: https://www.municipiobolivar.gob.ec/

GADM-Bolívar. (20 de Abril de 2015). *PLAN DE DESARROLLO Y ORDENAMIENTO TERRITORIAL DEL CANTÓN BOLÍVAR*. Obtenido de www.municipiobolivar.gob.ec: http://www.municipiobolivar.gob.ec/images/PDF/2015/04/pdot.pdf

Gama García, Á. M. (Junio de 2020). *Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles.* Obtenido de www.rua.ua.es: https://rua.ua.es/dspace/bitstream/10045/107576/1/Aplicacion_de_percepcion_mejorada_para_personas_con_Gama_Garcia_Angel_Manuel.pdf

Gandhi, R. (09 de Julio de 2018). *R-CNN, R-CNN rápido, R-CNN más rápido, YOLO - Algoritmos de detección de objetos [Imagen]*. Obtenido de www.towardsdatascience.com: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

González Muñiz, A. (01 de Febrero de 2018). *Aplicaciones de técnicas de inteligencia artificial basadas en aprendizaje profundo (deep learning) al análisis y mejora de la eficiencia de procesos industriales.* Obtenido de www.digibuo.uniovi.es:

http://digibuo.uniovi.es/dspace/bitstream/10651/45097/7/TFMAnaGonzalezMunizRUO.pdf

González Ulibarry, P. (Marzo de 2019). *Consecuencias ambientales de la aplicación de fertilizantes*. Obtenido de www.obtienearchivo.bcn.cl: https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio/10221/27059/1/Consecuencias_ambientales_de_la_aplicacion_de_fertilizantes.pdf

Ichi.Pro. (s.f.). *LeNet-5 en Kotlin con TensorFlow*. Obtenido de www.ichi.pro: https://ichi.pro/es/lenet-5-en-kotlin-con-tensorflow-202630331086503

Ichi.Pro. (s.f.). *Revisión de algoritmos de aprendizaje profundo para la detección de objetos [Imagen]*. Obtenido de www.ichi.pro: https://ichi.pro/it/revisione-degli-algoritmi-di-deep-learning-per-il-rilevamento-di-oggetti-213252981635154

infoAgro. (s.f.). *Manejo del Trips Occidental de las Flores* . Obtenido de www.infoagro.com: https://www.infoagro.com/hortalizas/trips.htm

Jiménez Silva, I. (2018). *Reconocimiento facial basado en redesneuronales convolucionales.* Obtenido de www.idus.us.es: https://idus.us.es/bitstream/handle/11441/85086/TFG-1975-JIMENEZ.pdf?sequence=1&isAllowed=y

Joyce Xu. (08 de Junio de 2018). *Uso del aprendizaje profundo para el reconocimiento de objetos*. Obtenido de www.deeplearningitalia.com: https://www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos/

Li, L., Zhang, S., & Wang, B. (08 de Abril de 2021). *Detección y clasificación de enfermedades de las plantas mediante aprendizaje profundo: una revisión*. Obtenido de https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/9399342

Luuk Heinsius. (18 de Junio de 2021). *Real-Time YOLOv4 FPGA Design with Catapult High-Level Synthesis*. Obtenido de www.essay.utwente.nl: http://essay.utwente.nl/86465/1/Heinsius_MA_EEMCS.pdf

Maeda Gutiérrez, V. (13 de Septiembre de 2019). *Comparación de arquitecturas de redes neuronales convolucionales para la clasificación de enfermedades en tomate.* Obtenido de www.ricaxcan.uaz.edu.mx: http://ricaxcan.uaz.edu.mx/jspui/bitstream/20.500.11845/1696/1/2019-09%20VALERIA%20MAEDA%20GTZ.pdf

MAGAP. (13 de Noviembre de 2018). *Boletin Situacional - Arveja 2017*. Obtenido de www.fliphtml5.com: https://fliphtml5.com/ijia/dkgk/basic

Manoharan , N. (30 de Abril de 2017). *Clasificador ImageNet con TensorFlow*. Obtenido de www.github.com: https://github.com/narenkmanoharan/ImageNet-Classifier-Tensorflow

Mohamed Elgendy. (2020). *Deep Learning for Vision Systems.* Manning Publications. Obtenido de www.livebook.manning.com: https://livebook.manning.com/book/grokking-deep-learning-for-computer-vision/chapter-7/v-5/1

Oliva Rodríguez, A. (2018). *Desarrollo de una aplicación de reconocimiento en imágenes utilizandoDeep Learningcon OpenCV*. Obtenido de www.riunet.upv.es: https://riunet.upv.es/bitstream/handle/10251/107243/OLIVA%20-%20Desarrollo%20de%20una%20aplicaci%C3%B3n%20de%20reconocimiento%20en%20im%C3%A1genes%20utilizando%20Deep%20Learning%20con%20O....pdf?sequence=1&isAllowed=y

Ordoñez Ramos, E. (2020). *Deep Learning para la visión artificial e identificación del personal administrativo y docente de la Universidad Nacional Micaela Bastidas de Apurímac 2018*. Obtenido de www.repositorio.unap.edu.pe: http://repositorio.unap.edu.pe/bitstream/handle/UNAP/13523/Erech_Ordo%C3%B1ez_Ramos.pdf?sequence=3&isAllowed=y

Paspuel Vera, O. J. (2013). *Evaluación de la adaptabilidad de cuatro variedades de arveja de tutoreo (Pisumsativum L.) Carchi –Ecuador*. Obtenido de www.repositorio.upec.edu.ec: http://repositorio.upec.edu.ec/bitstream/123456789/11/1/027%20EVALUACI%c3%93

N%20DE%20LA%20ADAPTABILIDAD%20DE%20CIATRO%20VARIEDADES%20DE%20ARVEJA%20DE%20TUTOREO%20%28PISUM%20SATIVUM%20L.%29%20CARCHI%20-%20ECUADOR%20O%20-%20PASPUEL%2c%20OLIVA.pdf

Patil, P., & Sachapara, V. (2018 de Febrero de 2018). *Providing smart agricultural solutions/techniques by using IoT based toolkit*. Obtenido de www.ieeexplore.ieee.org: https://ieeexplore.ieee.org/abstract/document/8300942

Pérez Lorenzo, C. (Junio de 2019). *Detección precoz de cáncer de piel en imágenes basado en redes convolucionales*. Obtenido de www.repositorio.uam.es: https://repositorio.uam.es/bitstream/handle/10486/688935/perez_lorenzo_cristina_tfg.pdf?sequence=1

Prado Jiménez, D. R. (2019). *Identificación de las principales plagas del cultivo de arveja (Pisum sativum L.)en la comunidad de Canchaguano, cantón Montúfar, provincia del Carchi.* Obtenido de www.dspace.utb.edu.ec: http://dspace.utb.edu.ec/bitstream/handle/49000/6873/E-UTB-FACIAG-ING%20AGRON-000205.pdf?sequence=1&isAllowed=y

ResearchGate. (Enero de 2019). *Capas y arquitecturas de redes neuronales convolucionales*. Obtenido de www.researchgate.net: https://www.researchgate.net/figure/VGGNet-architecture-19_fig2_333242381

Ruz Gómez, R., & Simón Rodríguez, A. (2020). *Evaluación de algoritmos de Machine Learning para la conducción*. Obtenido de https://eprints.ucm.es/: https://eprints.ucm.es/id/eprint/61910/1/Ruz_Gomez_Entrega_memoria_TFG_Evaluacion_de_algoritmos_para_la_conduccion_4398577_1208955137.pdf

Sánchez Arteaga, B. A. (2019). *Evaluación de un bioactivador de resistencias como tratamiento preventivo contra antracnosis Ascochyta pisi Lib.en arveja Pisum sativum L. de crecimiento indeterminado en el Centro Experimental San Francisco Carchi-Ecuador.* Obtenido de www.http://repositorio.upec.edu.ec: http://repositorio.upec.edu.ec/bitstream/123456789/903/1/371%20Evaluaci%c3%b3n%

20de%20un%20bioactivador%20de%20resistencias%20como%20tratamiento%20preventivo%20contra%20antracnosis.pdf

Sánchez Martínez, M. (Noviembre de 2018). *Detección de personas mediante técnicas de aprendizaje automático: SVM y CNN.* Obtenido de www.http://oa.upm.es: http://oa.upm.es/53880/1/TFG_MARINA_SANCHEZ_MARTINEZ.pdf

Sanz Cabreros, D. (17 de Febrero de 2019). *Desarrollo e implementación IoT de un sistema de reconocimiento de imágenes a nivel industrial*. Obtenido de www.reunir.unir.net: https://reunir.unir.net/bitstream/handle/123456789/8172/SANZ%20CABREROS%2C%20DAVID.pdf?sequence=1&isAllowed=y

Secretaría Técnica Planifica Ecuador. (2017). *Plan Nacional de Desarrollo 2017-2021*. Obtenido de www.planificacion.gob.ec: https://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL_0K.compressed1.pdf

Siddharth Das. (16 de Noviembre de 2017). *Analytics Vidhya[Imagen]*. Obtenido de www.medium.com: https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

slideplayer. (2016). *TABLA PARA VALORES DE CHI-CUADRADO CRÍTICO[Imagen]*. Obtenido de www.slideplayer.es: https://slideplayer.es/slide/10624268/35/images/17/TABLA+PARA+VALORES+DE+CHI-CUADRADO+CR%C3%8DTICO.jpg

Suntaxi Sarango, M. C. (27 de Marzo de 2019). *Detección de técnicas de aprendizaje profundo aplicadas en las diferentes áreas del conocimiento, empleando el método de revisión sistemática de literatura.* Obtenido de www.dspace.unl.edu.ec: https://dspace.unl.edu.ec/jspui/bitstream/123456789/21918/1/Suntaxi%20Sarango%20Martha%20Cristina.pdf

Tirira Tirira , E. J. (Marzo de 2018). *Estudio de factibilidad de un centro de acopio para la comercialización de arveja tierna (pisum sativum) en la provincia del Carchi*. Obtenido de www.repositorio.utn.edu.ec:

http://repositorio.utn.edu.ec/bitstream/123456789/8114/1/03%20AGN%20032%20TRABAJO%20DE%20GRADO.pdf

Toledo Perdomo , C. E., & Sagastume Mena, H. A. (Junio de 2018). *Diversidad de los tisanopteros (Insecta: Thysanoptera) presenta en el cultivo de arveja china (Pisum sativum L.), Santa Apolonia, Guatemala.* Obtenido de www.revistaespirales.com: http://www.revistaespirales.com/index.php/es/article/view/238/224

Torres Alonzo, A. (Enero de 2020). *Detección de frutas en árboles.* Obtenido de www.eprints.ucm.es: https://eprints.ucm.es/59518/1/TORRES_ALONSO_Deteccion_de_frutas_en_arboles_4398576_875953906%20%281%29.pdf

Utrera Burgal, J. (2018). *Deep Learning básico con Keras (Parte 4): ResNet[Imagen]*. Obtenido de www.jesusutrera.com: http://www.jesusutrera.com/articles/article07.html

Vasudevan, A., Kumar, D., & Bhuvaneswari, N. (29 de Diciembre de 2016). *Precision farming using unmanned aerial and ground vehicles*. Obtenido de www.ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/7801229

Verma, S., Chug, A., & Singh, A. P. (03 de Diciembre de 2018). *Prediction Models for Identification and Diagnosis of Tomato Plant Diseases*. Obtenido de https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/8554842

Vidushi , M. (25 de Febrero de 2021). *YOLOv3: Algoritmo de detección de objetos en tiempo real (¿Qué hay de nuevo?) [Imagen]*. Obtenido de www.viso.ai: https://viso.ai/deep-learning/yolov3-overview/

Villota Neira, P. N. (14 de Octubre de 2019). *Implementación de una estación prototipo con visión artificial, aplicado a la agricultura de precisión*. Obtenido de www.dspace.ucuenca.edu.ec: https://dspace.ucuenca.edu.ec/bitstream/123456789/33492/1/Trabajo%20de%20Titulaci%C3%B3n.pdf

Xu, J. (08 de Julio de 2018). *Uso del aprendizaje profundo para el reconocimiento de objetos*. Obtenido de www.deeplearningitalia.com: https://www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos/

yw0nam. (04 de Febrero de 2020). *Implemento DenseNet usando TF 2.0 [Imagen]*. Obtenido de www.github.com: https://github.com/yw0nam/DenseNet

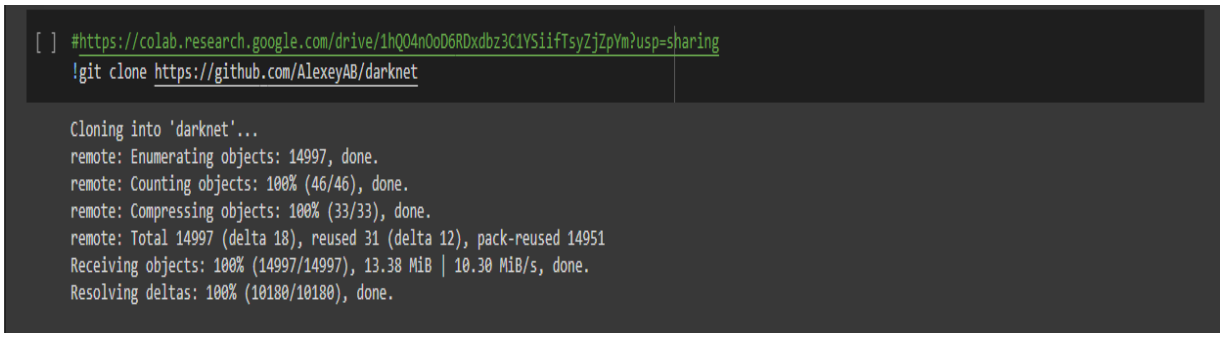
Zavala Salas, O. A. (Febrero de 2020). *Detección simultánea, por medio de CNN, de multiples robots NAO dentro de un campo de fútbol con apliación a RoboCup*. Obtenido de http://repositorio.utm.mx: http://repositorio.utm.mx/bitstream/123456789/303/1/2020-MR-OAZS.pdf

ANEXO A: ENTRENAMIENTO DE LA RED YOLOV4-TINY

Ingresar a Google Colab y seleccionar el acelerador de hardware GPU, a continuación, se procede a clonar la librería Darknet desde el repositorio GitHub AlexeyAB en la máquina virtual como se indica en la Figura 1.

Figura 1

Clonación de la librería Darknet desde el repositorio GitHub a la Google Colab



Nota: Clonación de la librería Darknet desde el repositorio GitHub a máquina virtual de Google Colab. Realizado por el Autor.

Configurar de la unidad Google Drive, para integrar la carpeta yolov4-tiny que contiene los archivos obj.names, obj.data, process.py, yolov4-tiny-custom.cfg y la carpeta backup a la máquina virtual como se muestra en la Figura 2.

Figura 2

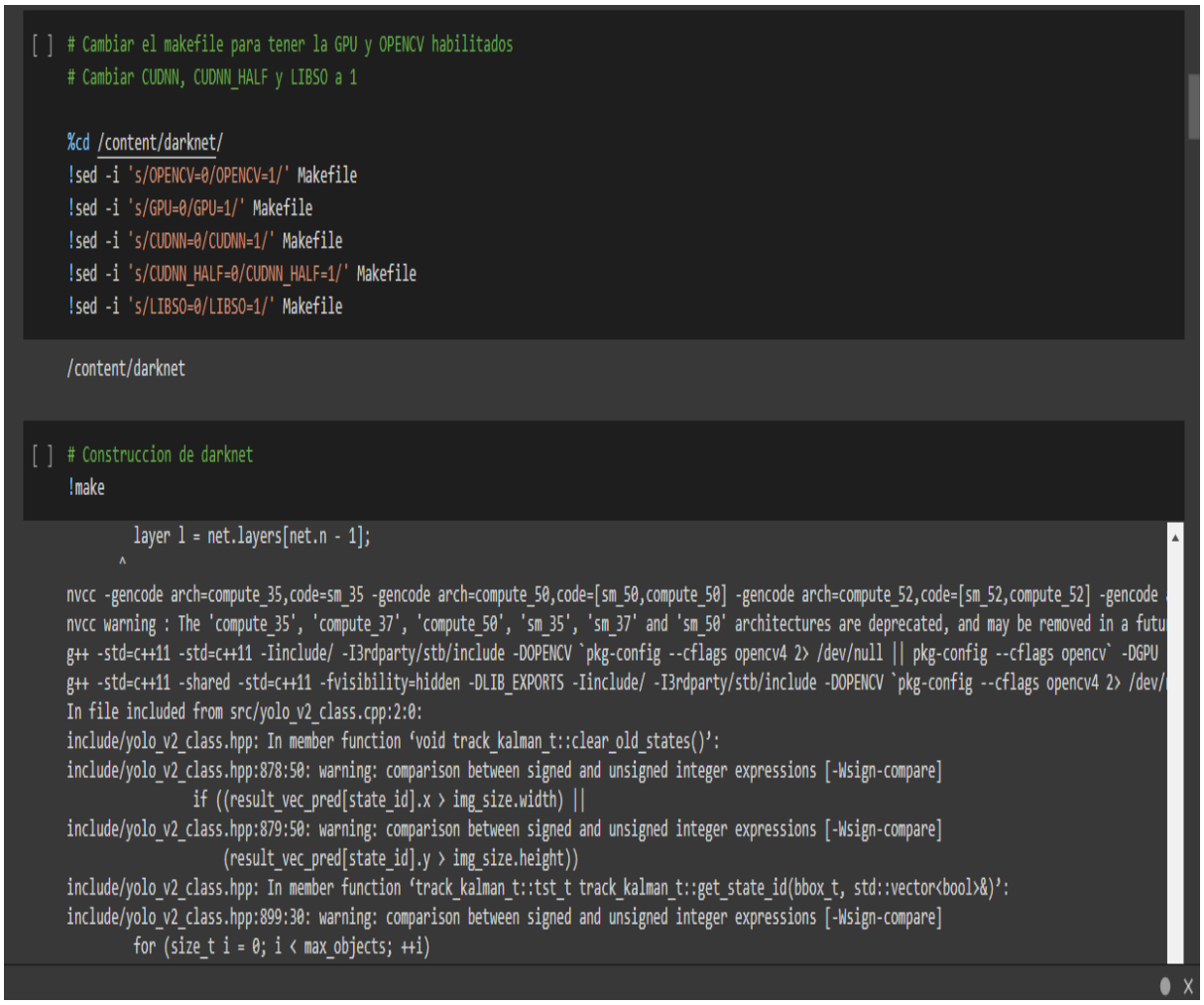
Integración de la carpeta yolov4-tiny a Google Colab

Nota: Integración de la carpeta yolov4-tiny a Google Colab. Realizado por el Autor.

Modificar el archivo Makefile para habilitar la operación de la GPU y OpenCV en la máquina virtual de Google Colab como se muestra en la Figura 3.

Figura 3

Modificación del archivo MAKE para habilitar OPENCV y GPU en Google Colab

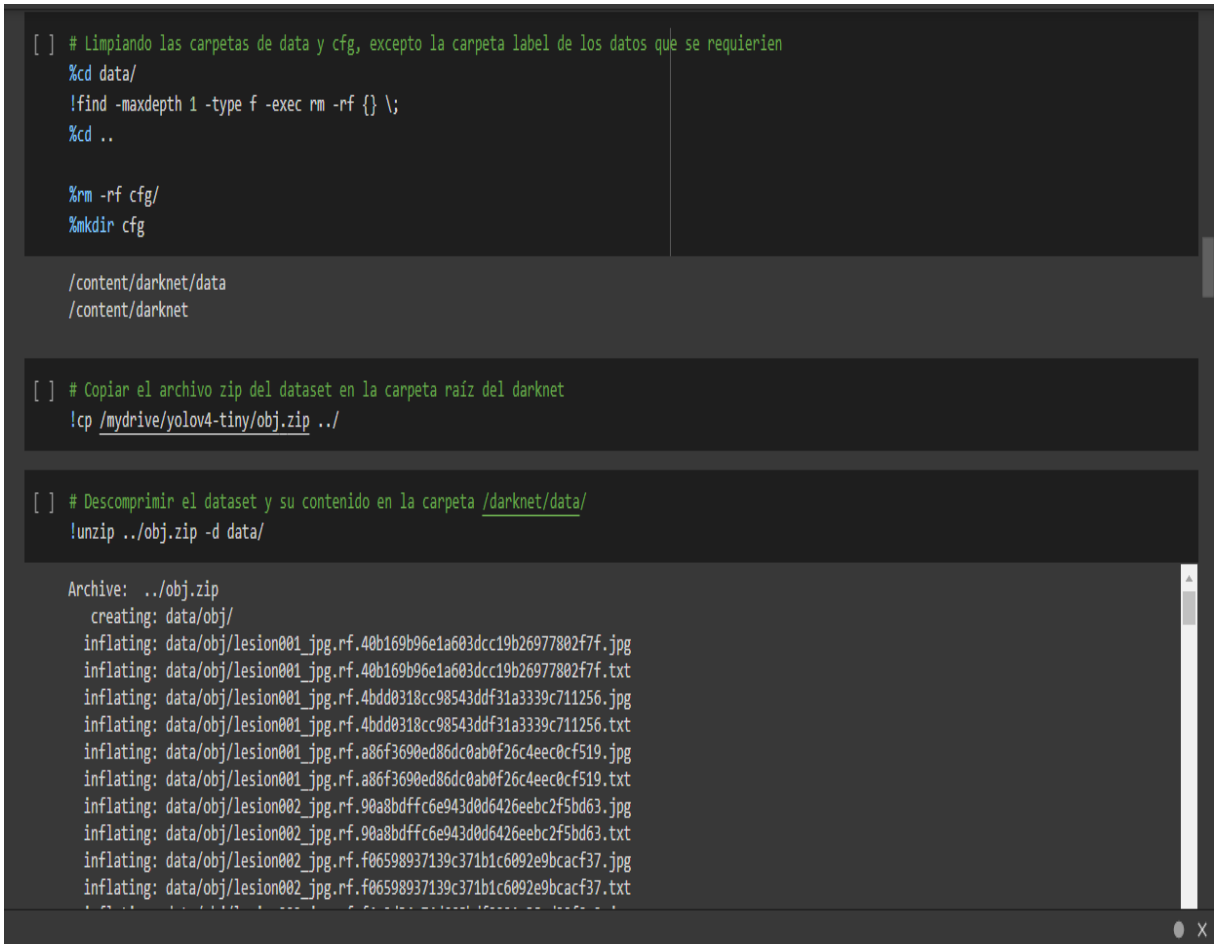


Nota: Modificación en el archivo MAKE para habilitar OPENCV y GPU en Google Colab. Realizado por el Autor.

Integración de los archivos del Dataset a la carpeta yolov4-tiny al directorio raíz Darknet en la máquina virtual de Google Colab.

Figura 4

Integración de los archivos del Dataset al directorio raíz Darknet en Google Colab



Nota: Copia de los archivos del Dataset al directorio raíz Darknet en Google Colab. Realizado por el Autor.

Copiar los archivos obj.names, obj.data de la carpeta yolov4-tiny al directorio Darknet/data de la máquina virtual de Google Colab como se muestra en la Figura 5.

Figura 5

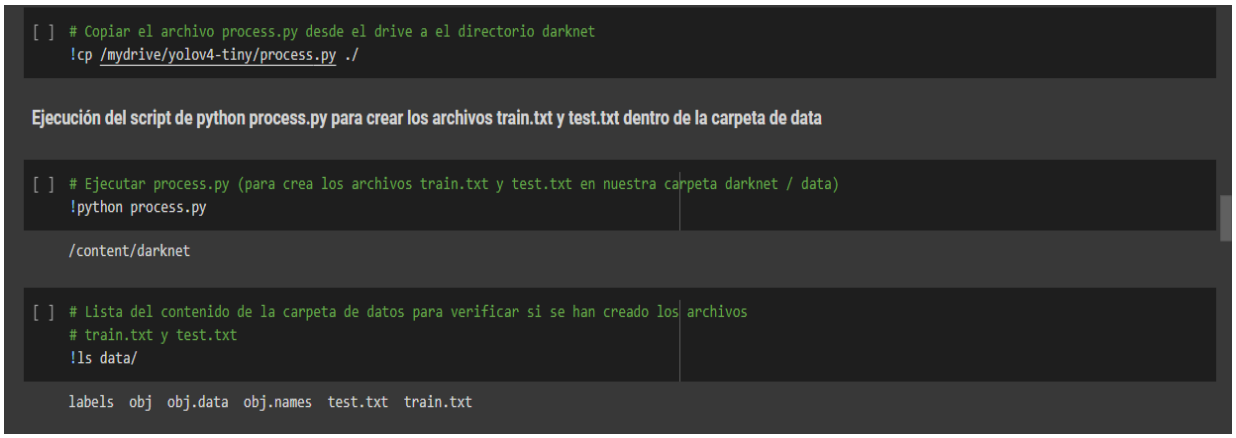
Copia de los archivos obj.names y obj.data al directorio raíz Darknet/data en Google Colab

Nota: Copia de los archivos obj.names y obj.data al directorio raíz Darknet/data en Google Colab. Realizado por el Autor.

Ejecutar el script process.py para la creación de los archivos train.txt y test.txt dentro de la carpeta de data como se indica en la Figura 6.

Figura 6

Creación de los archivos test.txt y train.txt al directorio Darknet/data en Google Colab

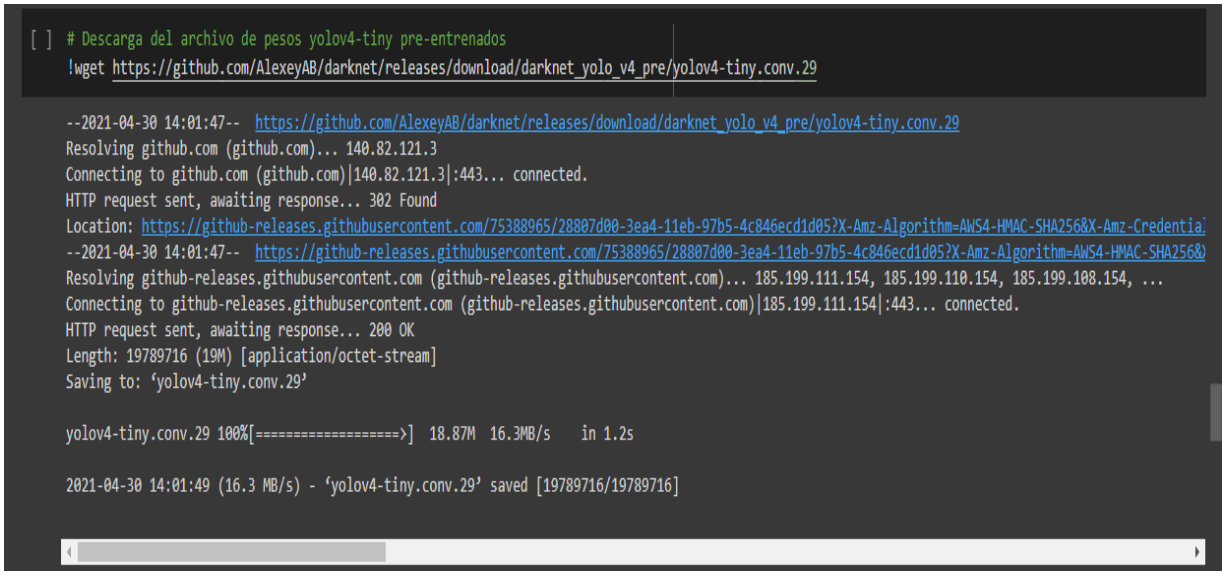


Nota: Creación de los archivos test.txt y train.txt al directorio Darknet/data en Google Colab. Realizado por el Autor.

Descargar los pesos pre-entrenados de la red neuronal yolov4-tiny.conv.29 desde el repositorio GitHub de AlexeyAB como se observa en la Figura 7.

Figura 7

Descarga de la red neuronal yolov4-tiny.conv.29 desde el repositorio GitHub AlexeyAB

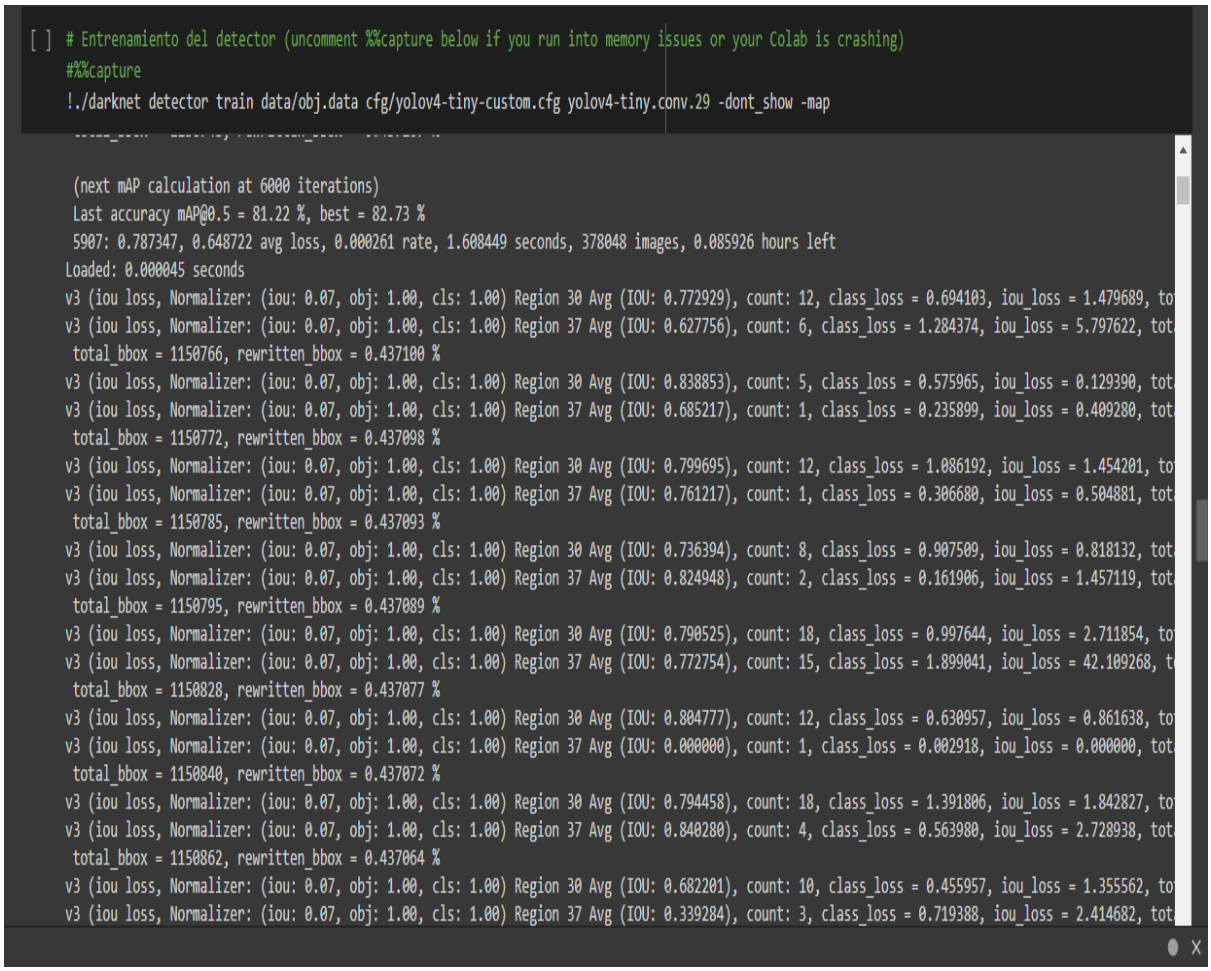


Nota: Descarga de la red neuronal yolov4-tiny.conv.29 desde el repositorio GitHub AlexeyAB. Realizado por el Autor.

Entrenar la red neuronal convolucional yolov4-tiny.conv.29 desde la máquina virtual de Google Colab como se indica en Figura 8.

Figura 8

Entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29 desde Google Colab

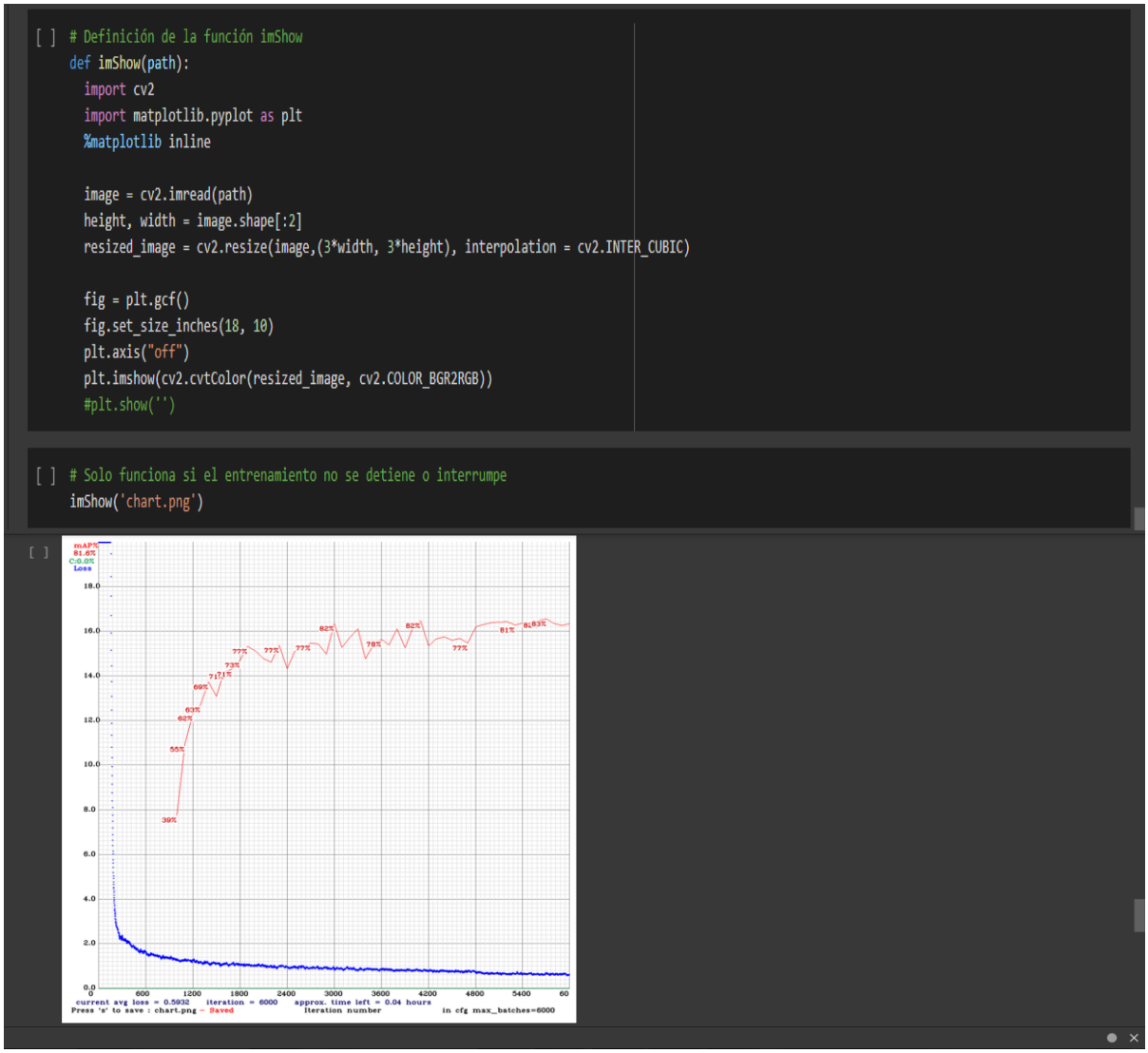


Nota: Entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29 desde Google Colab. Realizado por el Autor.

A continuación, se efectúa la comprobación del rendimiento de la red neuronal convolucional yolov4-tiny.conv.29 entrenada como se observa en la Figura 9.

Figura 9

Comprobación del rendimiento la red neuronal convolucional yolov4-tiny.conv.29



Nota: Rendimiento la red neuronal convolucional yolov4-tiny.conv.29 en la detección de objetos. Realizado por el Autor.

ANEXO B: ARCHIVOS PARA EL ENTRENAMIENTO DE LA RED

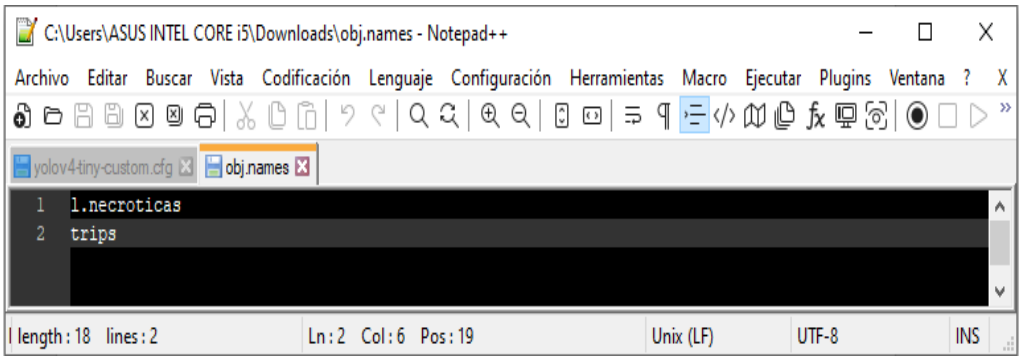
Para el entrenamiento de la red neuronal yolov4-tiny se necesita de los creación y configuración de los siguientes archivos:

- obj.names
- obj.data
- yolov4-tiny-custom.cfg
- train.txt
- valid.txt

El archivo obj.names, contiene el nombre de cada una de las clases de los objetos que serán entrenados como se muestra en la Figura 1.

Figura 1

Configuración del archivo obj.names



Nota: Configuración del archivo obj.names que se utilizará en el entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29 en la detección de objetos. Realizado por el Autor.

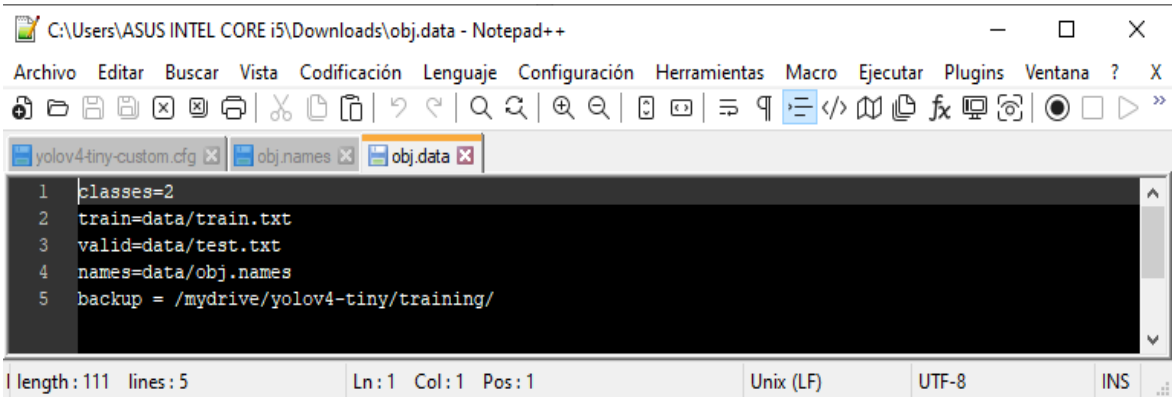
Para la configuración del archivo obj.data, es necesario crear los archivos train.txt y valid.txt. El archivo process.py permite generar los archivos train.txt y valid.txt con una distribución de 90-10 respectivamente del total de imágenes del dataset.

- El archivo train.txt contiene la ruta de todas las imágenes del dataset de entrenamiento.

- El archivo valid.txt contiene la ruta de todas las imágenes del dataset de validación.
- En el archivo obj.data se configuran las rutas de los archivos necesarios para el entrenamiento.

Figura 2

Configuración del archivo obj.data



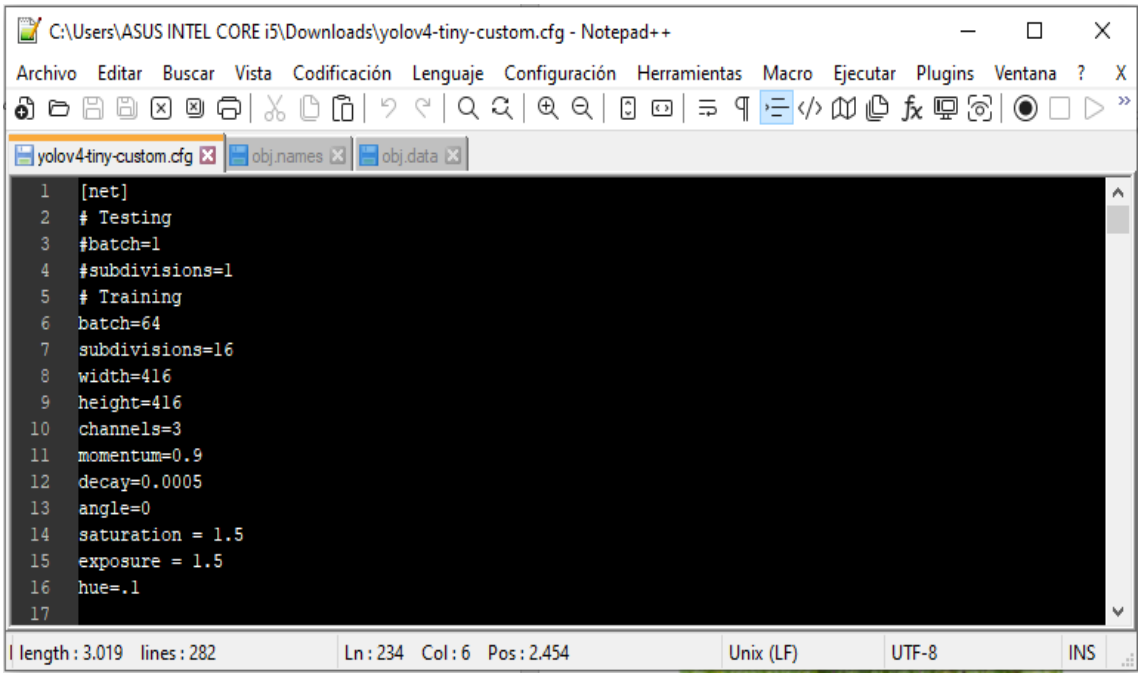
Nota: Configuración del archivo obj.data que se utilizará en el entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29 en la detección de objetos. Realizado por el Autor.

En el archivo yolov4-tiny-custom.cfg, se configura la arquitectura neuronal de yolov4-tiny.

- Línea 6: batch=64 #Configuración dependiendo de la tarjeta gráfica
- Línea 7: subdivisions=16 #Configuración dependiendo de la tarjeta gráfica
- Línea 212: filters=21 #(clases + 5)*3 Número de filtros
- Línea 220: classes=2 #Número de clases de entrenamiento
- Línea 263: filters=21 #(clases + 5)*3 Número de filtros
- Línea 269: classes=2 #Número de clases de entrenamiento

Figura 3

Configuración del archivo yolov4-tiny-custom.cfg



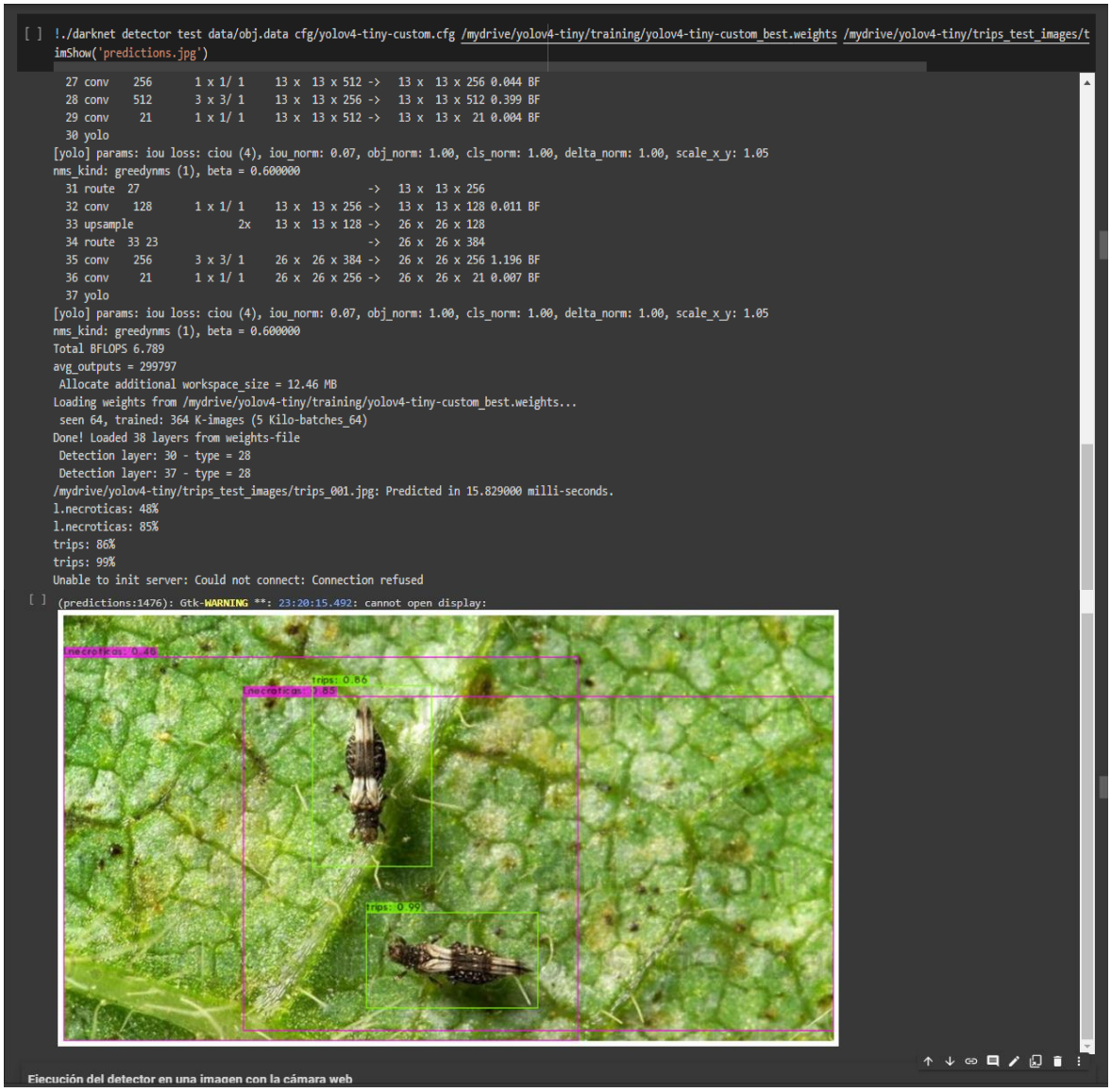
Nota: Configuración del archivo yolov4-tiny-custom.cfg que se utilizará en el entrenamiento de la red neuronal convolucional yolov4-tiny.conv.29 en la detección de objetos. Realizado por el Autor.

ANEXO C: PRUEBAS DE LA RED YOLOV4-TINY

Ejecutando la detección de la plaga thrips en una imagen de prueba con la red entrenado yolov4-tiny.conv.29.

Figura 1

Detección de la plaga thrips con la red yolov4-tiny.conv.29



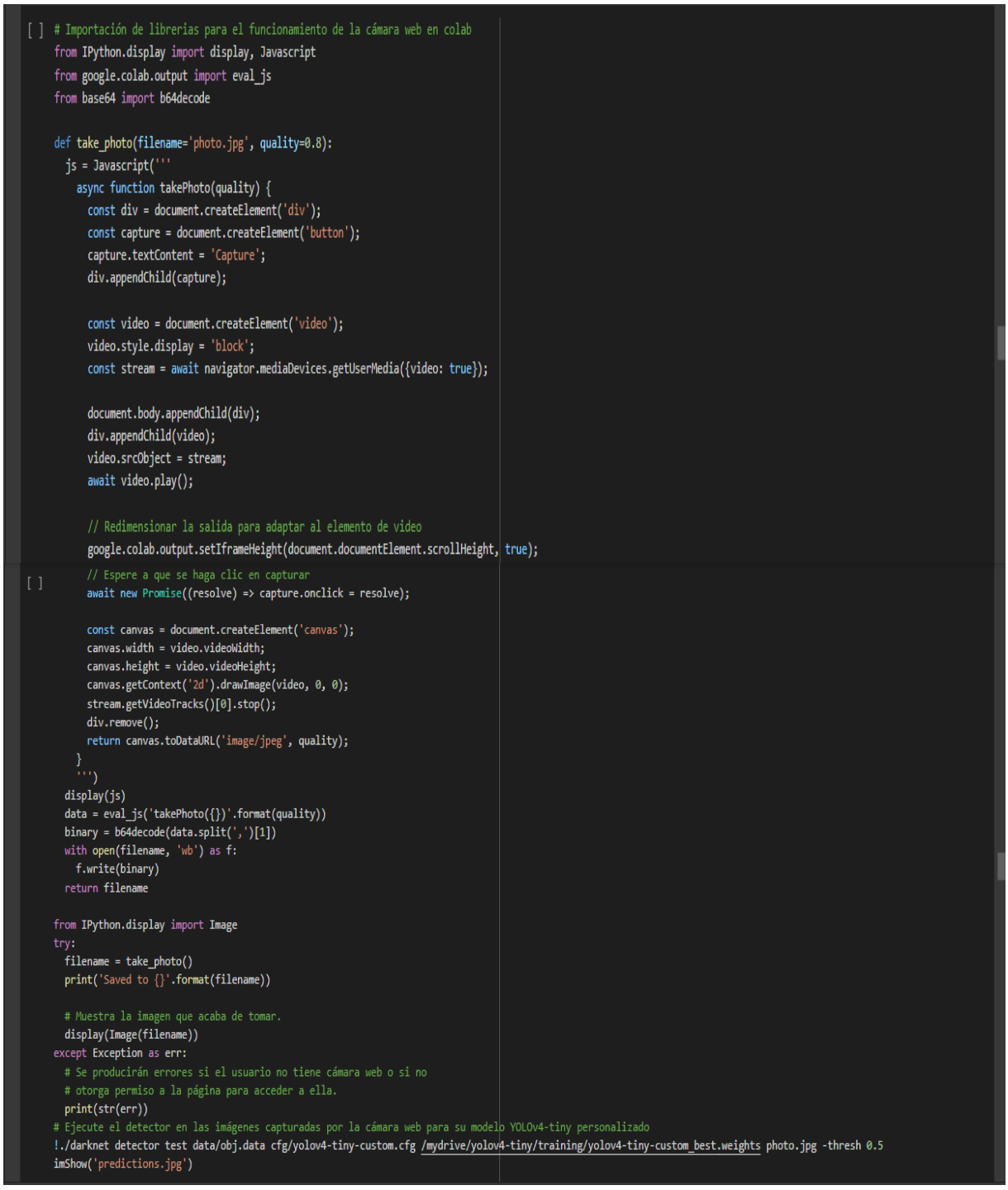
Nota: Detección de la plaga thrips con la red yolov4-tiny.conv.29 en imágenes. Realizado por el Autor.

Ejecutando la detección de la plaga thrips con la red entrenada yolov4-tiny.conv.29, en una imagen que fue capturada desde una cámara web, en la Figura 2 se

indica el código que fue utilizado para el funcionamiento de la cámara web en Google Colab.

Figura 2

Código utilizado para el funcionamiento de la cámara web en Google Colab

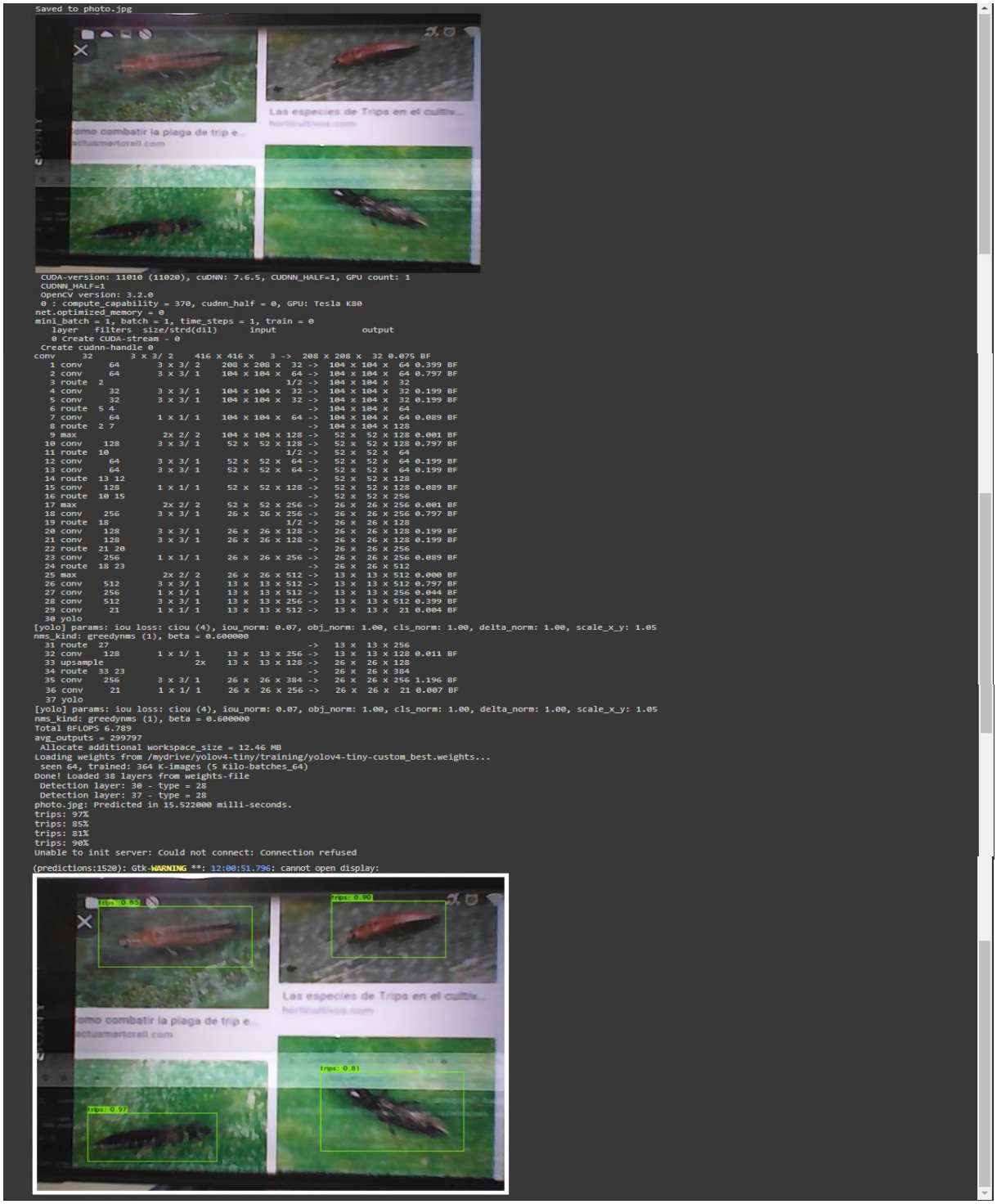


```
[ ] # Importación de librerias para el funcionamiento de la cámara web en colab
    from IPython.display import display, Javascript
    from google.colab.output import eval_js
    from base64 import b64decode

    def take_photo(filename='photo.jpg', quality=0.8):
      js = Javascript('''
        async function takePhoto(quality) {
          const div = document.createElement('div');
          const capture = document.createElement('button');
          capture.textContent = 'Capture';
          div.appendChild(capture);

          const video = document.createElement('video');
          video.style.display = 'block';
          const stream = await navigator.mediaDevices.getUserMedia({video: true});

          document.body.appendChild(div);
          div.appendChild(video);
          video.srcObject = stream;
          await video.play();

          // Redimensionar la salida para adaptar al elemento de video
          google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);
[ ]       // Espere a que se haga clic en capturar
          await new Promise((resolve) => capture.onclick = resolve);

          const canvas = document.createElement('canvas');
          canvas.width = video.videoWidth;
          canvas.height = video.videoHeight;
          canvas.getContext('2d').drawImage(video, 0, 0);
          stream.getVideoTracks()[0].stop();
          div.remove();
          return canvas.toDataURL('image/jpeg', quality);
        }
        ''')
      display(js)
      data = eval_js('takePhoto({})'.format(quality))
      binary = b64decode(data.split(',')[1])
      with open(filename, 'wb') as f:
        f.write(binary)
      return filename

    from IPython.display import Image
    try:
      filename = take_photo()
      print('Saved to {}'.format(filename))

      # Muestra la imagen que acaba de tomar.
      display(Image(filename))
    except Exception as err:
      # Se producirán errores si el usuario no tiene cámara web o si no
      # otorga permiso a la página para acceder a ella.
      print(str(err))
    # Ejecute el detector en las imágenes capturadas por la cámara web para su modelo YOLOv4-tiny personalizado
    !./darknet detector test data/obj.data cfg/yolov4-tiny-custom.cfg /mydrive/yolov4-tiny/training/yolov4-tiny-custom_best.weights photo.jpg -thresh 0.5
    imShow('predictions.jpg')
```

Nota: Código utilizado para el funcionamiento de la cámara web en Google Colab para la detección de la plaga thrips. Realizado por el Autor.

En la Figura 3 se observa la detección de la plaga thrips en una imagen capturada desde una cámara web.

Figura 3

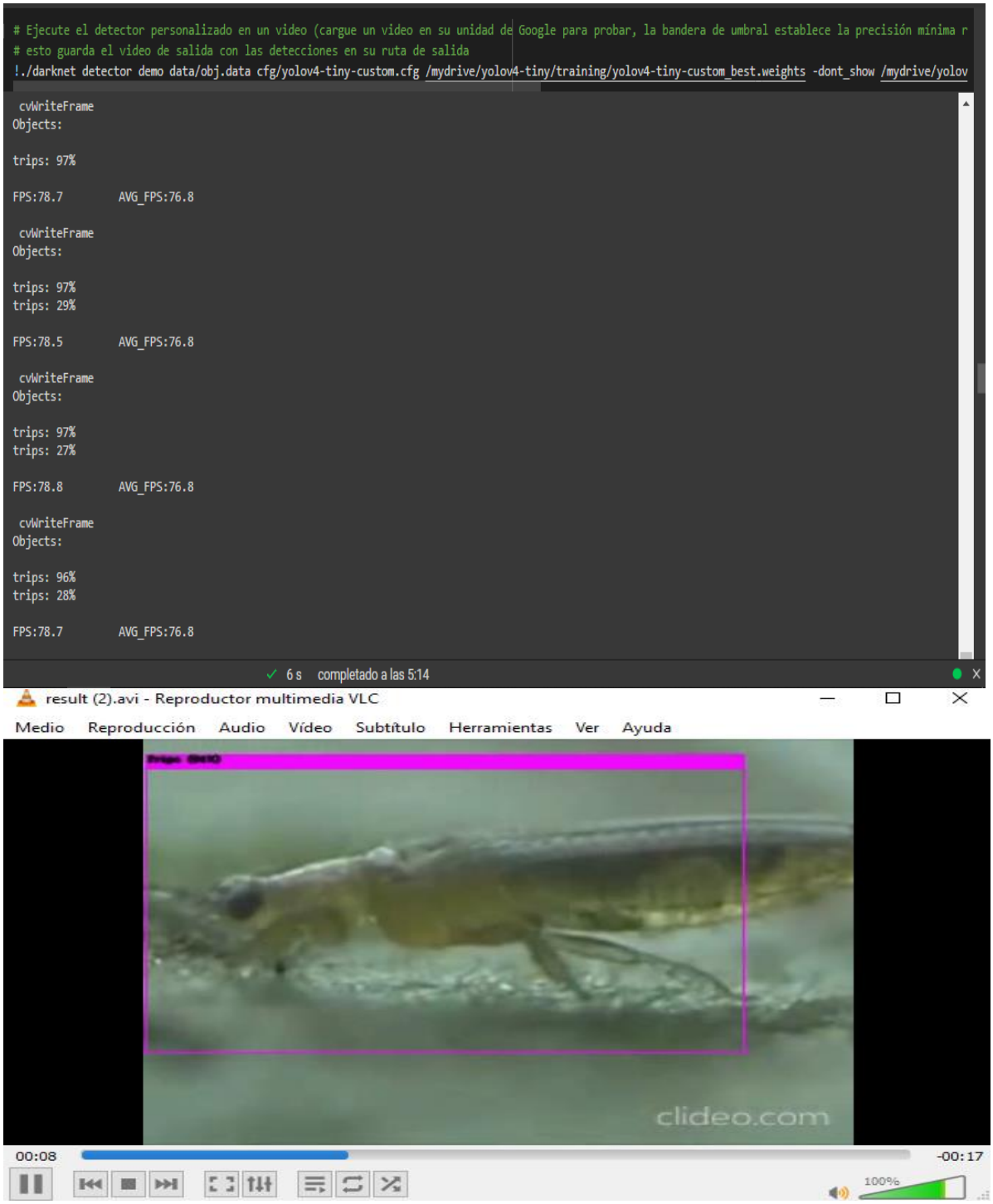


Nota: Detección de la plaga thrips en una imagen capturada por una cámara web en tiempo real. Realizado por el Autor.

Finalmente, en la Figura 4 se realiza la detección de la plaga thrips en un video con la red entrenada yolov4-tiny.conv.29.

Figura 4

Detección de la plaga thrips en un video con la red yolov4-tiny.conv.29



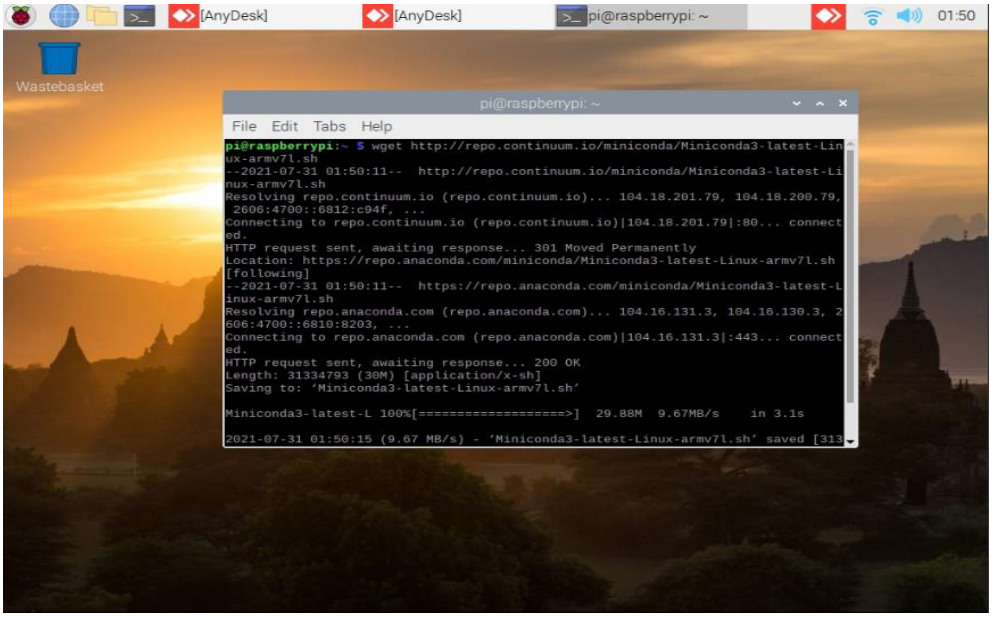
Nota: Detección de la plaga thrips en un video con la red yolov4-tiny.conv.29. Realizado por el Autor.

ANEXO D: INSTALACIÓN DE PAQUETES Y LIBRERIAS

En la Figura 1 se indica la descarga del paquete anaconda a la SBC Raspberry-Pi, desde los repositorios http://repo.continuum.io/miniconda/Miniconda3-latest-Linux-armv7l.sh.

Figura 1

Descarga del paquete anaconda a Raspberry-Pi

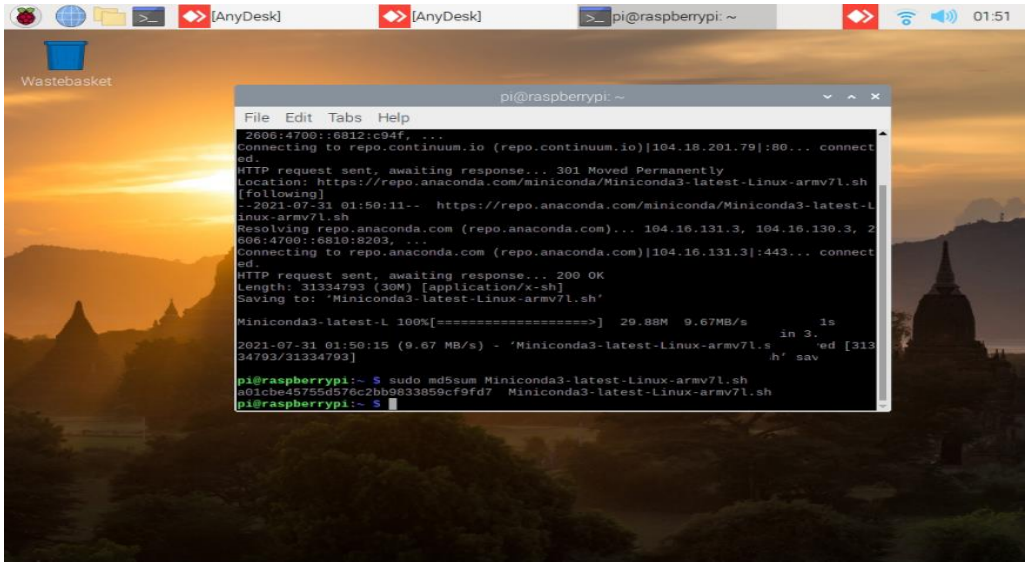


Nota: Descarga del paquete anaconda para la SBC Raspberry-Pi desde los repositorios. Realizado por el Autor.

A continuación, se inicia la instalación del paquete anaconda en la SBC Raspberry-Pi como se observa en la Figura 2.

Figura 2

Inicio de la instalación del paquete anaconda en Raspberry-Pi

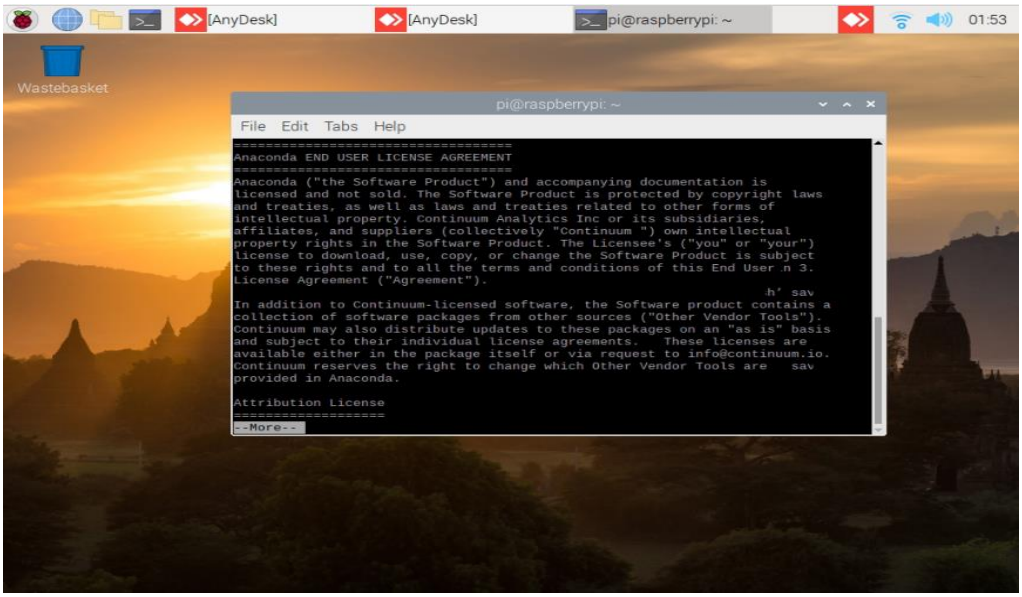


Nota: Instalación del paquete anaconda en Raspberry-Pi. Realizado por el Autor.

En seguida, se proceda a aceptar la licencia del paquete anaconda para continuar con la instalación como se indica en la Figura 3.

Figura 3

Admisión de licencia del paquete anaconda

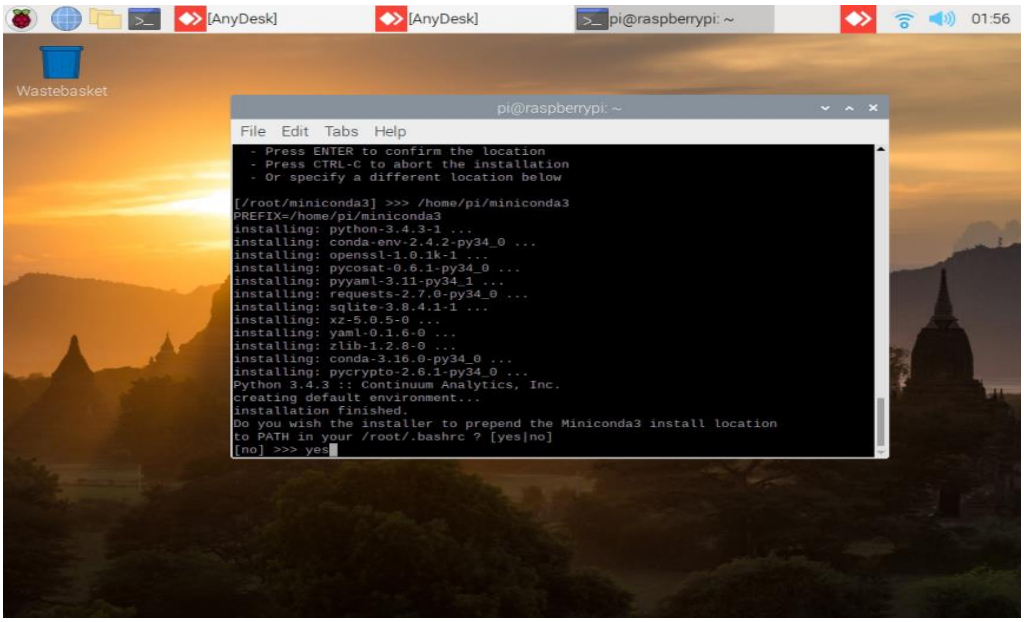


Nota: Admisión de la licencia del paquete anaconda en Raspberry-Pi. Realizado por el Autor.

Una vez aceptada la licencia se realiza la configuración de la ruta, /home/pi/miniconda3 donde serán instalados los paquetes o librerías de anaconda como se observa en la Figura 4.

Figura 4

Configuración de la ruta para la instalación de los paquetes de anaconda.

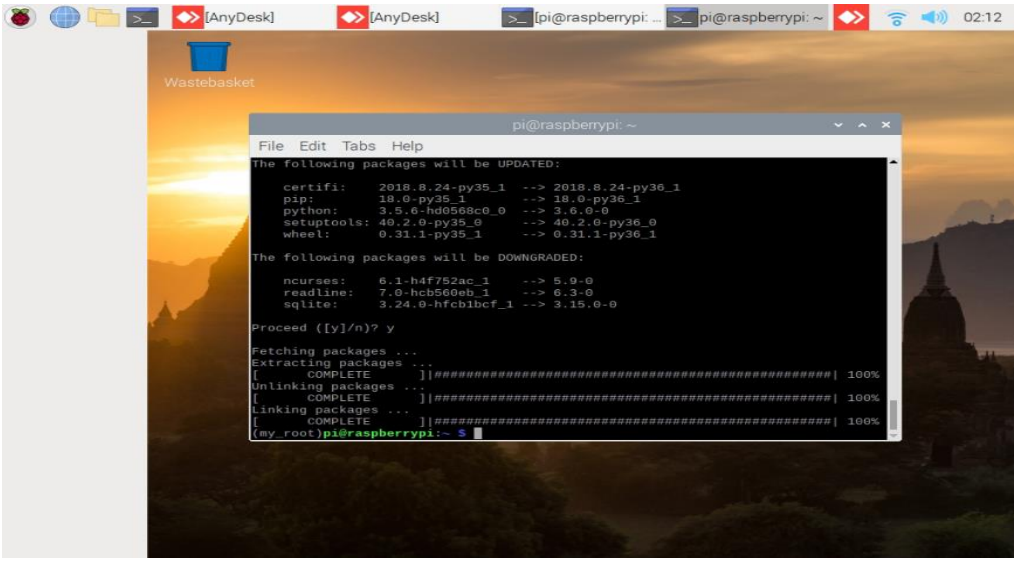


Nota: Configuración la ruta donde se realizará la instalación de todos los paquetes de anaconda en la Raspberry-Pi. Realizado por el Autor.

Después, se efectúa la actualización de la herramienta Python 3.7.7 y la instalación de librerías complementarias de Python, como se indica en la Figura 5. Para realiza esta tarea se tiene que abrir a una nueva terminal de Raspberry-Pi e ingresar las siguientes líneas.

Figura 5

Actualización e instalación del paquete Python 3.7

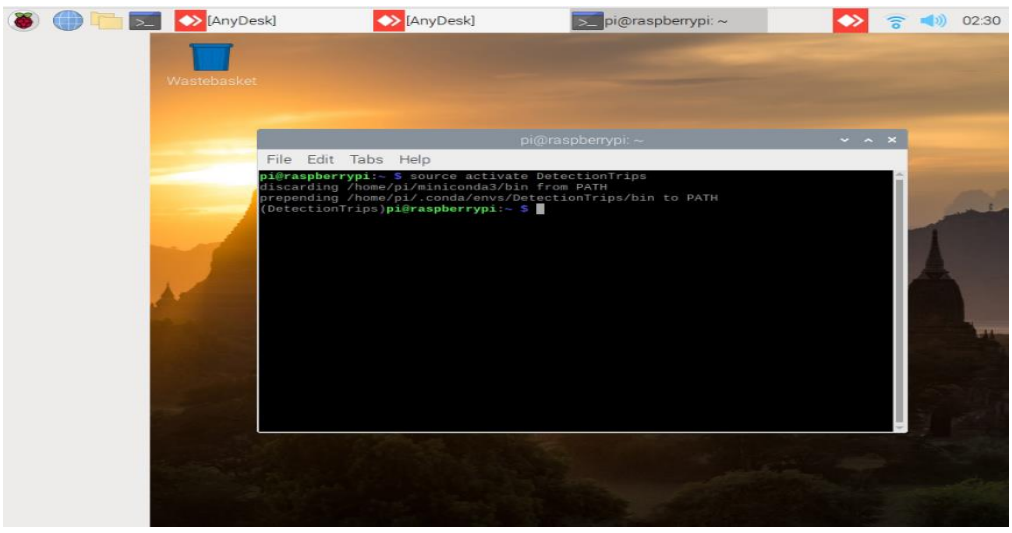


Nota: Actualización y la instalación del paquete Python 3.7 y librerías complementarias desde la terminal de Raspberry-Pi. Realizado por el Autor.

Una vez instalado el paquete Python 3.7.7, se procede a crear un entorno virtual como se observa en la Figura 6. Para la instalación de los programas y librerías que serán empleados en el desarrollo del sistema de reconocimiento de lesiones necróticas para la detección temprana de la plaga thrips en los cultivos del guisante o arveja.

Figura 6

Creación del entorno virtual DetectionThrips

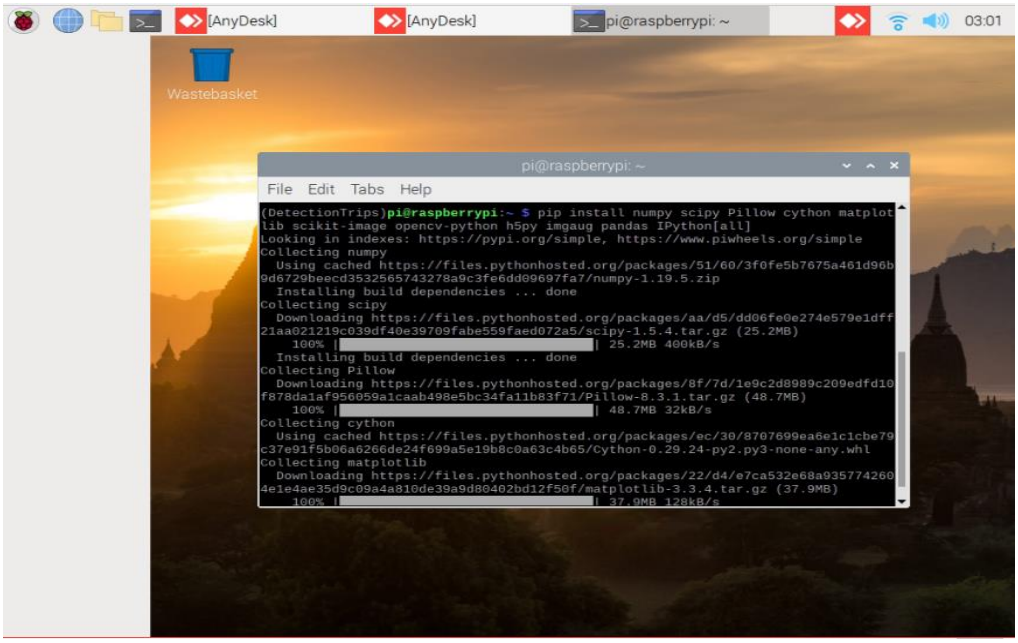


Nota: Creación del entorno virtual DetectionThrips para la instalación de paquetes y librerías que serán empleados en el desarrollo del sistema. Realizado por el Autor.

Finalmente, en la Figura 7 se indica la instalación de los programas y librerías que serán utilizadas en el desarrollo del sistema propuesto.

Figura 7

Instalación de programas y librerías para el desarrollo del sistema propuesto



Nota: Instalación de los programas y librerías que se utilizarán para el desarrollo del sistema propuesto dentro del entorno virtual DetectionThrips. Realizado por el Autor.

ANEXO E: DESARROLLO DE LA APP WEB

En la Figura 1 se muestra la interfaz principal de la App Web, donde se observa el nombre de la aplicación, una descripción de la App y un menú para seleccionar la tarea o acción que se efectuará.

Figura 1

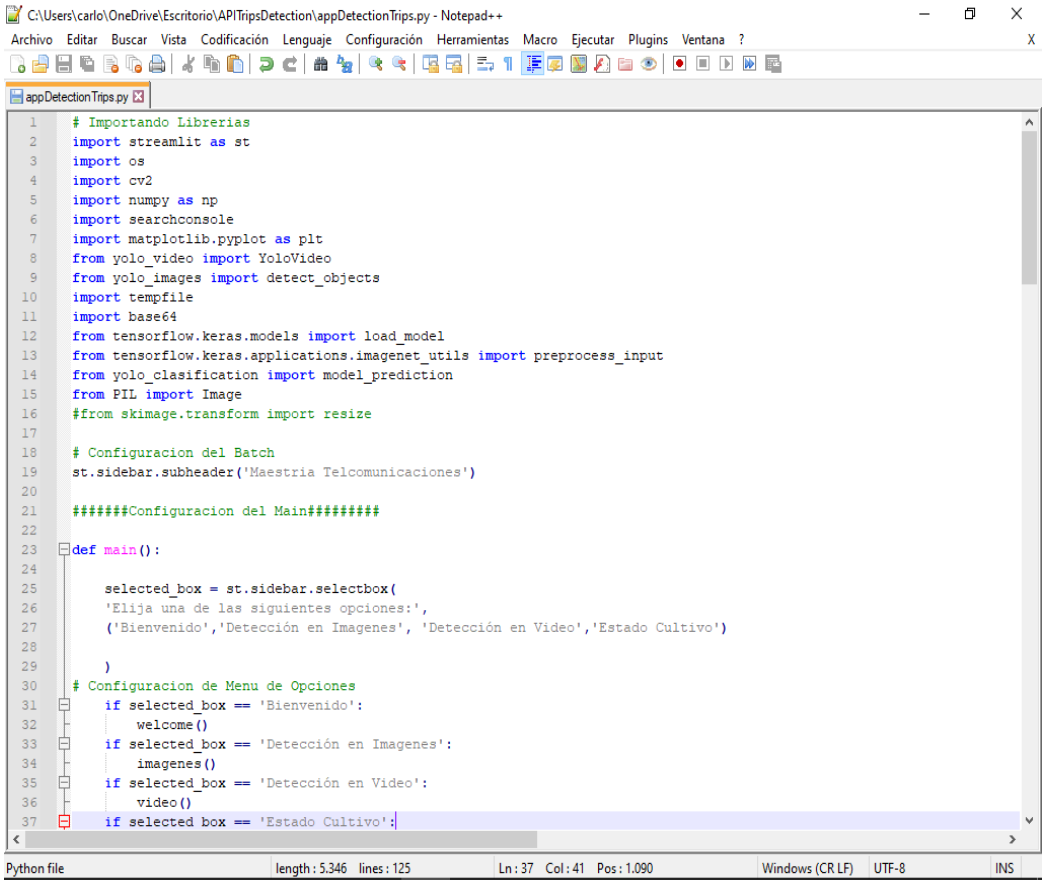
Entorno de la interfaz principal de la App Web

Nota: Interfaz principal de la App desarrollada para la detección de la plaga thrips en el guisante o arveja. Realizado por el Autor.

A continuación, en la Figura 2 se indica el código utilizado en el desarrollo de interfaz principal de la App Web.

Figura 2

Código utilizado para interfaz principal de la App Web

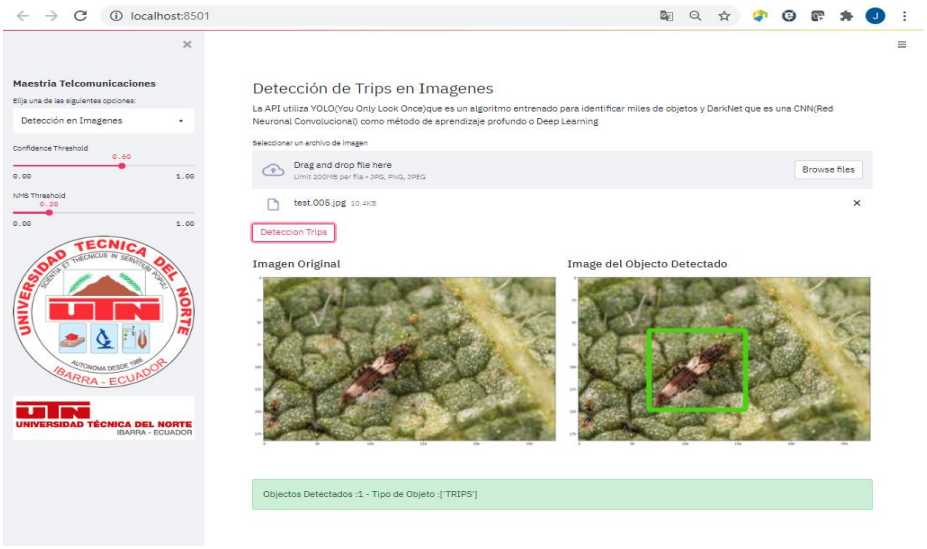


Nota: Código utilizado en la interfaz principal de la App Web desarrollada para la ejecución de la tarea o acción a efectuarse. Realizado por el Autor.

En la Figura 3 se indica la interfaz web de la detección de la plaga thrips en el guisante o arveja en imágenes digitales. Donde se observar la imagen original (izquierda) y la imagen procesada (derecha), además muestra información sobre la cantidad de objetos detectados y el tipo de objeto que es detectado.

Figura 3

Detección de la plaga thrips en imágenes con la App Web

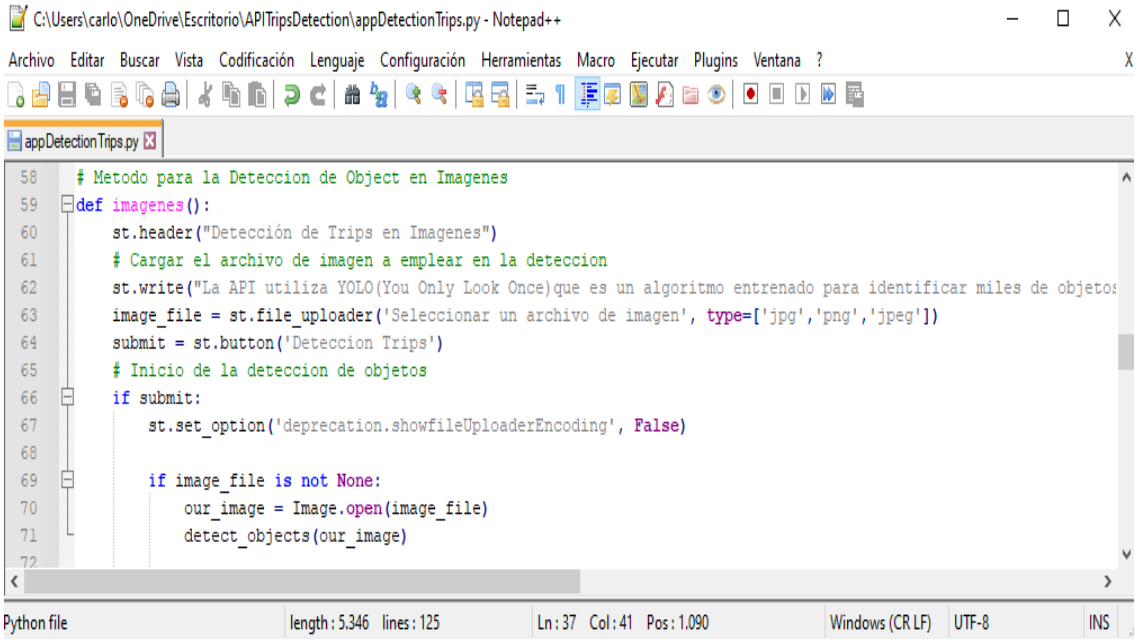


Nota: Interfaz de la App Web desarrollada para la detección de la plaga thrips en el guisante o arveja en imágenes. Realizado por el Autor.

A continuación, en la Figura 4 se muestra el código utilizado para la detección de la plaga thrips en imágenes.

Figura 4

Código utilizado para la detección de la plaga thrips con la App Web en imágenes

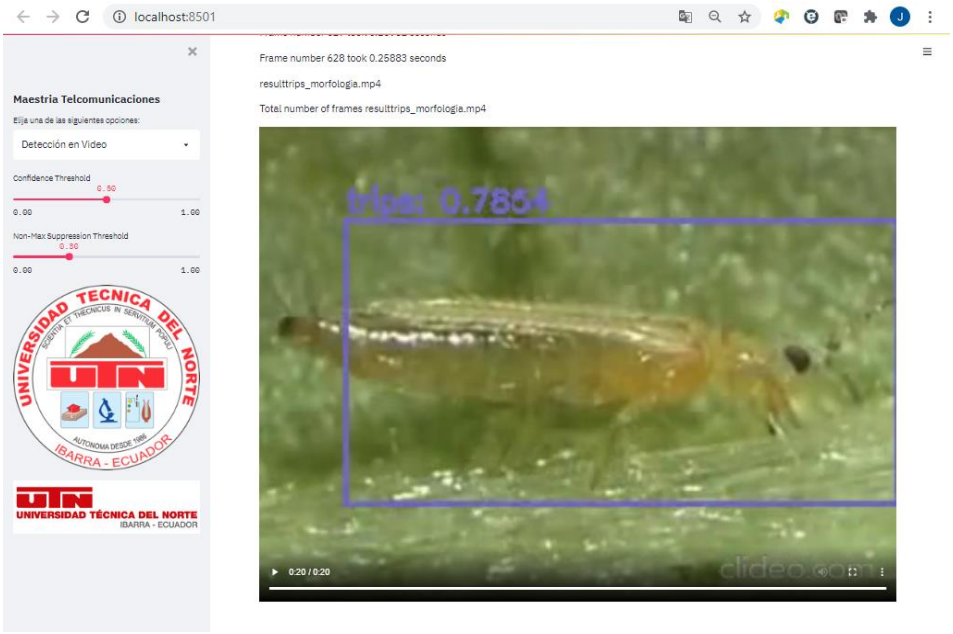


Nota: Código utilizado para la detección de la plaga thrips con la App Web en imágenes. Realizado por el Autor.

En la Figura 5 se observa la interfaz web de la detección de la plaga thrips en el guisante o arveja en videos.

Figura 5

Detección de la plaga thrips en videos con la App Web

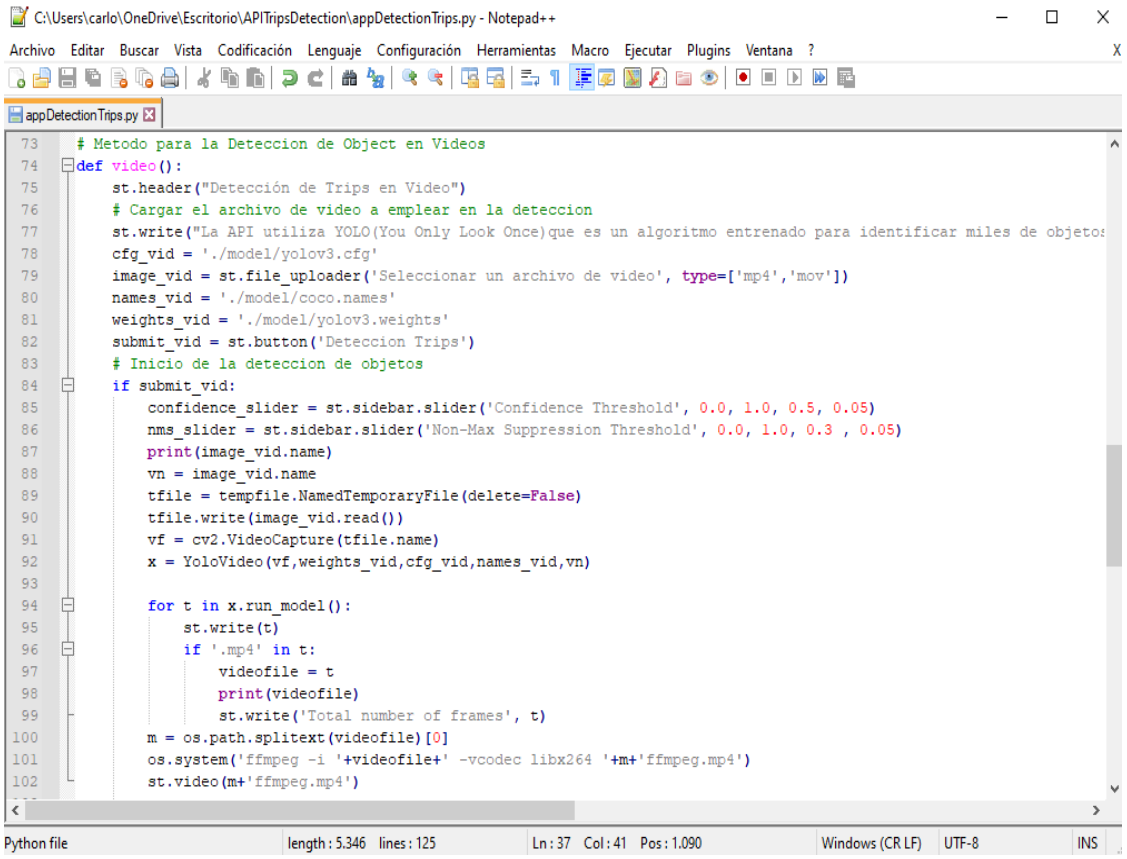


Nota: Interfaz de la App desarrollada para la detección de la plaga thrips en el guisante o arveja en videos. Realizado por el Autor.

En la Figura 6 se indica el código utilizado para la detección de la plaga thrips en video.

Figura 6

Código utilizado para la detección de la plaga thrips con la App Web en videos

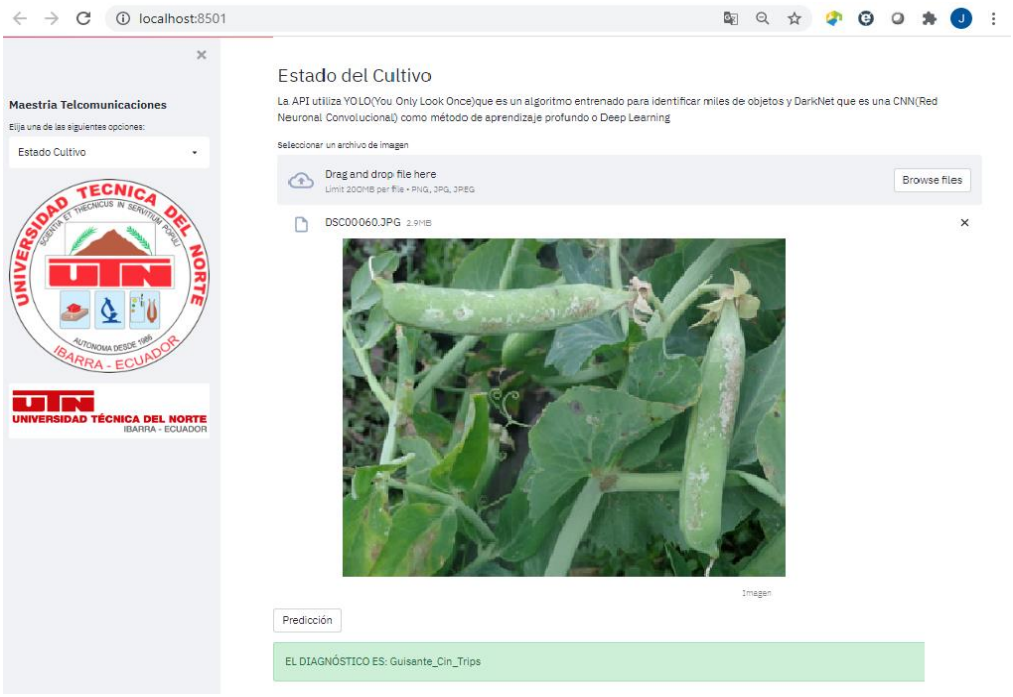


Nota: Código utilizado para la detección de la plaga thrips con la App Web en videos. Realizado por el Autor

En la Figura 7 se observa la interfaz web de la App Web que realiza el pronóstico de estado actual del guisante o arveja.

Figura 7

Pronóstico del estado actual del cultivo con la App Web

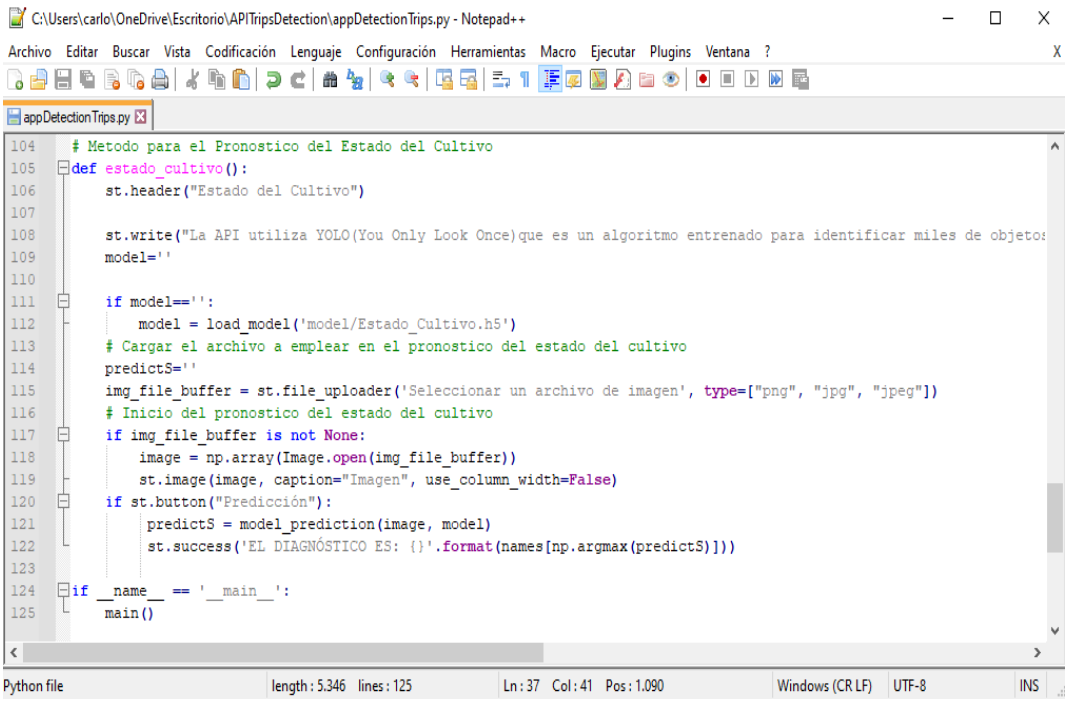


Nota: Interfaz de la App Web desarrollada para el pronóstico del estado actual del cultivo. Realizado por el Autor.

A continuación, en la Figura 8 se indica el código utilizado para del pronóstico del estado actual del cultivo del guisante o arveja.

Figura 8

Código utilizado para la predicción del estado actual del cultivo con la App Web



Nota: Código utilizado para la predicción del estado actual del cultivo del guisante o arveja con la App Web. Realizado por el Autor.