

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Software

Detección automática de malezas en cultivos de maíz y/o papa mediante imágenes adquiridas por un dron, basado en la arquitectura de red neuronal convolucional VGG.

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado
ante la ilustre Universidad Técnica del Norte

Autor:

Edwin Sebastian Cabascango Salazar

Director:

PhD. Iván Danilo García Santillán

Ibarra – Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1003879101
APELLIDOS Y NOMBRES:	Cabascango Salazar Edwin Sebastian
DIRECCIÓN:	
EMAIL:	escabascango@utn.edu.ec
TELÉFONO FIJO:	----
TELÉFONO MÓVIL:	0969518074

DATOS DE LA OBRA	
TÍTULO:	Detección automática de malezas en cultivos de maíz y/o papa mediante imágenes adquiridas por un dron, basado en la arquitectura de red neuronal convolucional VGG.
AUTOR (ES):	Edwin Sebastian Cabascango Salazar
FECHA DE APROBACIÓN: DD/MM/AAAA	30/07/2024
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero de software
ASESOR /DIRECTOR:	PhD. Iván Danilo García Santillán

2. CONSTANCIAS

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

Ibarra, a los 30 días del mes de julio de 2024

EL AUTOR:

Edwin Sebastian Cabascango Salazar
1003879101

CERTIFICACIÓN DIRECTOR

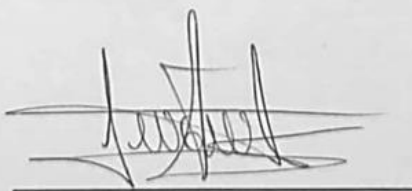
Ibarra, 30 de julio del 2024

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo PhD. Iván García, certifico que el sr. Edwin Sebastian Cabascango Salazar portador de la cedula de ciudadanía número 1003879101, ha trabajado en el desarrollo del proyecto de grado **“Detección automática de malezas en cultivos de maíz y/o papa mediante imágenes adquiridas por un dron, basado en la arquitectura de red neuronal convolucional VGG.”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente



PhD. Iván Danilo García Santillán
DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

Con el corazón lleno de gratitud y emoción, dedico esta tesis:

A mi madre, pilar inquebrantable de mi vida. Tu amor incondicional y apoyo constante han sido la luz que ha iluminado cada paso de mi camino académico y personal. Sin tu presencia, este logro habría sido un sueño lejano. Hoy, se materializa gracias a ti. Te amo, mamá.

A mis hermanos, José, Humberto, Fernando, Alexander y Koya. Cada uno, a su manera única, ha contribuido a forjar mi carácter y determinación. Su apoyo firme ha sido un bálsamo en los momentos difíciles y una alegría en los triunfos.

A Dios, fuente inagotable de sabiduría y fortaleza. En los momentos de duda, tu presencia me recordó que todo es posible para quien tiene fe.

Y a mí mismo, por la perseverancia demostrada ante cada obstáculo. Este trabajo es testimonio de que los sueños se alcanzan con esfuerzo, dedicación y la voluntad de levantarse después de cada caída.

AGRADECIMIENTOS

Al concluir esta etapa trascendental de mi vida académica, expreso mi más profunda gratitud:

A mi madre Carmencita, ejemplo vivo de amor y sacrificio. Tu noble corazón y tu incansable lucha por brindarme educación son el cimiento de este logro. Tus palabras de aliento y tu amor incondicional han sido el motor que me impulsó a no rendirme jamás.

A mis amigos Fabián, Felipe, Adrián, Andrés y Saya, por su apoyo inquebrantable y su capacidad para motivarme en los momentos más críticos. Nuestra amistad ha sido un regalo invaluable en este viaje.

A Linda por ayudarme a encontrarme y ser una mejor persona.

A Alex y Johao, compañeros de carrera y amigos entrañables. Su presencia hasta el final de este camino ha sido fundamental. Juntos hemos crecido, aprendido y superado cada desafío.

Al PhD. Iván García, mi tutor y guía. Su sabiduría y dirección han sido cruciales para dar forma y profundidad a esta investigación. Su mentoría trasciende lo académico y deja una huella imborrable en mi formación profesional.

Al PhD. Marco Pusedá, por su ayuda esencial en la recolección de imágenes para este proyecto de investigación y por compartir consejos de vida que han enriquecido no solo este trabajo, sino mi perspectiva personal.

No puedo dejar de reconocer a aquellos cuya influencia, aunque sutil, fue decisiva en mi camino. A cada persona que, con un gesto, una palabra o una acción, me impulsó a perseverar en este proyecto, les extiendo mi más sincero agradecimiento. Este logro no es solo mío, sino el resultado de una sinergia de esfuerzos, apoyo y amor. Que este trabajo sea un testimonio de que, con determinación y el respaldo de quienes nos aman, podemos alcanzar las estrellas.

TABLA DE CONTENIDOS

DEDICATORIA.....	4
AGRADECIMIENTOS.....	5
TABLA DE CONTENIDOS.....	6
ÍNDICE DE FIGURAS.....	10
ÍNDICE DE TABLAS.....	11
RESUMEN.....	12
ABSTRACT	14
INTRODUCCIÓN	16
Objetivos.....	17
<i>Objetivo General</i>	17
Objetivos Específicos.....	17
Alcance.....	18
Metodología	20
Justificación	21
CAPÍTULO I.....	24
Marco Teórico.....	24
1.1 Fundamentación Teórica	24
1.1.1 Malezas.....	24
1.1.2 Control de malezas.....	25
1.1.3 Tipos de malezas en cultivo de maíz	26
1.1.4 Tipos de malezas en cultivos de papa	27
1.1.5 Imágenes por dron	28
1.1.6 Vehículos aéreos no tripulados (Drones)	29
1.2 Redes Neuronales Convolucionales.....	29

1.2.1 Inteligencia Artificial (IA)	29
1.2.2 Vision Artificial.....	30
1.2.3 Redes neuronales convolucionales (CNN).....	30
1.2.4 Componentes, arquitectura y técnicas de entrenamiento	32
1.2.5 Aprendizaje supervisado y no supervisado	35
1.2.6 Herramientas y bibliotecas para trabajar con redes neuronales convolucionales	36
1.2.7 Red Neuronal Convolucional VGG.....	38
1.2.8 Características de la arquitectura VGG	39
1.3 Normas y Metodologías de Desarrollo	42
1.3.1 Proceso Interindustrial para minería de Datos (CRISP-DM).....	42
1.3.2 Fases de CRISP-DM.....	43
1.3.3 Metodología de desarrollo Kanban.....	47
1.3.4 Aplicación Web.....	48
1.3.5 Norma ISO 25010	49
1.4 Artículos Relacionados	50
1.5 Preguntas de Investigación.....	53
Capítulo 2.....	59
Desarrollo del proyecto	59
2.1 Metodologías y herramientas.....	59
2.1.1 Metodología CRISP-DM	59
2.1.2 Metodología kanban	61
2.1.3 Características del Dron del Dron.....	62
2.1.4 Herramientas de desarrollo	64
2.2 Construcción del conjunto de imágenes	66
2.2.1 Recopilación de imágenes.....	66

2.2.2	Preprocesamiento del dataset	74
2.2.3	Etiquetado de datos	76
2.3.4	Métricas del conjunto de datos final (Ojo revisar total de imágenes en RF)	80
2.3	Modelo de detección de objetos basado en VGG	81
2.4	Implementación y Evaluación del Modelo	83
2.4.1	Configuración de hiperparámetros	86
2.4.2	Entrenamiento del modelo	87
2.5	Implementación del modelo en entorno Web	89
2.5.1	Evaluación según estándar 25010	90
Capítulo 3	92
Resultados y validaciones	92
3.1	Métricas del modelo SSDVGG	92
3.1.1	Fases de entrenamiento	92
3.2	Comparación del modelo con métodos tradicionales.....	98
3.2.1	Inferencias y resultados	98
3.2.2	Métodos tradicionales.....	101
3.2.3	Uso del modelo de detección de malezas con Dron	101
3.3	Pruebas estadísticas	104
3.3.1	T – Student en tiempos de detección de malezas.....	104
3.3.2	T – Student en detección de malezas con etiquetas verdaderas y pagina web.....	106
3.3.3	T – Student en detección de papas con etiquetas verdaderas vs página web	109
3.3.4	T – Student en detección de plantas con etiquetas reales vs página web.....	112
DISCUSIÓN	115
CONCLUSIONES	117
RECOMENDACIONES	118

BIBLIOGRAFÍA 120

ÍNDICE DE FIGURAS

Figura 1 Árbol de problemas	17
Figura 2 Arquitectura de red neuronal	19
Figura 3 Síntesis del desarrollo del proyecto	21
Figura 4 Fases de CRISP-DM.	46
Figura 5 Despliegue del modelo en Gradio.....	49
Figura 6 Proceso CRISDM del proyecto.....	61
Figura 7 Tablero Kanban para gestión de proyectos.....	61
Figura 8 Dron DJI Mavic 2 Pro	63
Figura 9 Recursos Colab Pro	64
Figura 10 Modelo de detección en producción.....	65
Figura 11 Planta de papa a 4 semanas de crecimiento.....	67
Figura 12 Plantación de papa.....	67
Figura 13 Trayectoria y configuración del proyecto	68
Figura 14 información de imagen original.	74
Figura 15 Obstrucciones visibles en la fotografía.....	75
Figura 16 Imagen con altura que excede el límite establecido.....	75
Figura 17 Total de imágenes a etiquetar.	77
Figura 18 Etiquetas de los elementos a detectar.	77
Figura 19 Detalle final luego del preprocesamiento y aumento de datos en Roboflow.....	78
Figura 20 Formato Pascal VOC para entrenamiento del modelo.....	79
Figura 21 Detalles del archivo de etiquetas XML.....	80
Figura 22 Información descriptiva del conjunto de datos	80
Figura 23 Estado del conjunto de datos pertenecientes al Dataset #2 "Balanceado".	81
Figura 24 Arquitectura VGG.....	81
Figura 25 Arquitectura SSD con backbone VGG	82
Figura 26 Estructura de la carpeta "train" que contiene las imágenes junto con sus etiquetas.....	83
Figura 27 Recursos computacionales en Kaggle.....	84
Figura 28 Estructura de archivos kaggle.....	85
Figura 29 Estructura de archivos en Google Colab.....	86
Figura 30 Configuración de hiperparámetros.....	87
Figura 31 Reetiquetado a dos clases para entrenamiento del modelo.....	88
Figura 32 Integración del modelo en entorno web.....	89
Figura 33 Loss del entrenamiento y validación del modelo final.....	94
Figura 34 Mean IOU del entrenamiento y validación del modelo final.....	95
Figura 35 Mean Average Precision del entrenamiento y validación del modelo final.....	95

ÍNDICE DE TABLAS

Tabla 1 Malezas en cultivos de maíz.....	26
Tabla 2 Malezas en cultivo de papa.....	28
Tabla 3 Preguntas de investigación.....	53
Tabla 4 Especificaciones del Dron.....	62
Tabla 5 Detalles de las herramientas de desarrollo.....	65
Tabla 6 Fechas en las que se capturaron las imágenes en los terrenos elegidos.....	69
Tabla 7 Planta y malezas de la presente investigación.....	70
Tabla 8 Número de imágenes capturadas por fechas en los terrenos.....	71
Tabla 9 Parámetros elegidos para la captura de imágenes.....	73
Tabla 10 Banco de imágenes depuradas.....	76
Tabla 11 Experimentación en el entrenamiento del modelo.....	88
Tabla 12 Fases de entrenamiento.....	92
Tabla 13 IOU de cada clase de todas las versiones del modelo entrenadas utilizando el conjunto de validación.....	97
Tabla 14 Prueba 1 Etiquetas reales vs Predicciones del modelo.....	98
Tabla 15 Prueba 2. Etiquetas reales vs Predicciones del modelo.....	99
Tabla 16 Prueba 3. Etiquetas reales vs Predicciones del modelo.....	100
Tabla 17 Tiempo de detección entre método tradicional e inferencia del modelo.....	102
Tabla 18 Meticas de tiempos de detección de malezas.....	105
Tabla 19 Prueba T de Student.....	105
Tabla 20 Correlación de Pearson.....	105
Tabla 21 Resultado comparativo entre malezas con etiquetas verdaderas y pagina web.....	106
Tabla 22 Estadística de malezas con etiquetas verdaderas y pagina web.....	108
Tabla 23 Comparación entre detección de papas con etiquetas verdaderas vs página web.....	109
Tabla 24 Estadística de papas con etiquetas verdaderas y pagina web.....	110
Tabla 25 Detección de plantas con etiquetas reales vs página web.....	112
Tabla 26 Valores de mAP, IoU de trabajos relacionados.....	116

RESUMEN

La gestión eficaz de malezas es un componente crítico en las prácticas agrícolas modernas. Los agricultores deben implementar estrategias de control precisas para mitigar los efectos adversos de estas plantas invasoras en sus cultivos. Las malezas compiten agresivamente por recursos esenciales como nutrientes, agua, espacio y luz solar, lo que puede resultar en una disminución significativa tanto en la productividad como en la calidad nutricional de los cultivos.

El proceso de identificación de malezas presenta desafíos considerables, particularmente en términos de eficiencia temporal. La magnitud del tiempo requerido para esta tarea está directamente correlacionada con la extensión del área cultivada. Adicionalmente, factores como la experiencia del agricultor en reconocimiento de especies de malezas, la topografía del terreno, y las condiciones climáticas, influyen en la complejidad y duración del proceso. Estas variables a menudo resultan en evaluaciones imprecisas o estimaciones subjetivas, lo que puede llevar a decisiones subóptimas en las estrategias de control de malezas.

En respuesta a esta problemática, se propone el desarrollo de una aplicación web que incorpore algoritmos de aprendizaje profundo para la detección automatizada de malezas. Esta solución tecnológica utilizará imágenes capturadas por vehículos aéreos no tripulados (drones) como fuente primaria de datos. El objetivo principal es optimizar significativamente el tiempo dedicado a esta tarea crucial, proporcionando un análisis objetivo y preciso. Esta innovación promete mejorar la toma de decisiones en el manejo de malezas, contribuyendo así a una agricultura más eficiente y sostenible.

En el capítulo 1 se examina las diferencias fundamentales entre cultivos y malezas, categorizando las diversas especies de malezas. Se explora la aplicabilidad del aprendizaje profundo en la identificación de malezas, revisando las arquitecturas algorítmicas contemporáneas y los desafíos actuales en este

campo. Se establece un marco comparativo con investigaciones precedentes para contextualizar el estudio.

El capítulo 2 detalla el proceso de adquisición de datos y justifica la selección de SSD con red base VGG como algoritmo principal. Se describe la evolución y optimización del algoritmo para mejorar la precisión predictiva. Además, se explica la implementación de la interfaz web mediante Gradio y se detallan las metodologías empleadas a lo largo del proceso de desarrollo.

En el capítulo 3 se dedica a la evaluación comparativa de las diferentes iteraciones del algoritmo, culminando en la selección del modelo más eficaz para su integración en la aplicación web. La eficiencia de la solución tecnológica se evalúa utilizando los criterios establecidos por la norma ISO 25010. Se emplea el análisis estadístico t de Student para validar la mejora en la eficiencia del proceso de identificación y cuantificación de malezas facilitado por la aplicación.

Por último, en la sección de discusión se contrasta los hallazgos del estudio con investigaciones anteriores, evaluando el grado de innovación y el impacto potencial de esta nueva herramienta en el ámbito de la gestión agrícola moderna.

ABSTRACT

Effective weed management is a pivotal component in contemporary agricultural practices. Farmers must implement precise control strategies to mitigate the adverse effects of these invasive plants on their crops. Weeds aggressively compete for essential resources such as nutrients, water, space, and sunlight, potentially resulting in a significant decline in both crop productivity and nutritional quality.

The process of weed identification presents considerable challenges, particularly in terms of temporal efficiency. The magnitude of time required for this task directly correlates with the extent of the cultivated area. Additionally, factors such as the farmer's expertise in recognizing weed species, terrain topography, and climatic conditions influence the complexity and duration of the process. These variables often lead to imprecise assessments or subjective estimations, potentially resulting in suboptimal decisions regarding weed control strategies.

In response to this issue, we propose the development of a web application that incorporates deep learning algorithms for automated weed detection. This technological solution will utilize images captured by unmanned aerial vehicles (drones) as the primary data source. The main objective is to significantly optimize the time dedicated to this crucial task, providing an objective and accurate analysis. This innovation promises to enhance decision-making in weed management, thus contributing to more efficient and sustainable agricultural practices.

In the first chapter 1 examines the fundamental differences between crops and weeds, categorizing various weed species. It explores the applicability of deep learning in weed identification, reviewing contemporary algorithmic architectures and current challenges in this field. A comparative framework with preceding research is established to contextualize the study.

In the second chapter details the data acquisition process and justifies the selection of SSD with VGG base network as the primary algorithm. It describes the evolution and optimization of the algorithm to improve predictive accuracy. Furthermore, it explains the implementation of the web interface using Gradio and outlines the methodologies employed throughout the development process.

The third chapter is devoted to the comparative evaluation of various algorithm iterations, culminating in the selection of the most effective model for integration into the web application. The efficiency of the technological solution is assessed using criteria established by the ISO 25010 standard. Student's t-statistical analysis is employed to validate the enhancement in efficiency of the weed identification and quantification process facilitated by the application.

Lastly, the discussion section contrasts the study's findings with previous research, evaluating the degree of innovation and the potential impact of this new tool in the field of modern agricultural management.

INTRODUCCIÓN

El problema de la detección de malezas en cultivos es un desafío importante en la industria agrícola, ya que las malezas pueden competir con las plantas cultivadas por nutrientes, agua y luz, lo que puede reducir la producción y la calidad de los cultivos. Una forma de abordar este problema es mediante el uso de imágenes obtenidas por drones, que pueden proporcionar una visión aérea de los cultivos con alta resolución y precisión. Sin embargo, el proceso manual de examinar estas imágenes puede ser tedioso y propenso a errores humanos. Por lo tanto, es importante desarrollar métodos automatizados para detectar y clasificar malezas en imágenes de drones de manera precisa y eficiente.

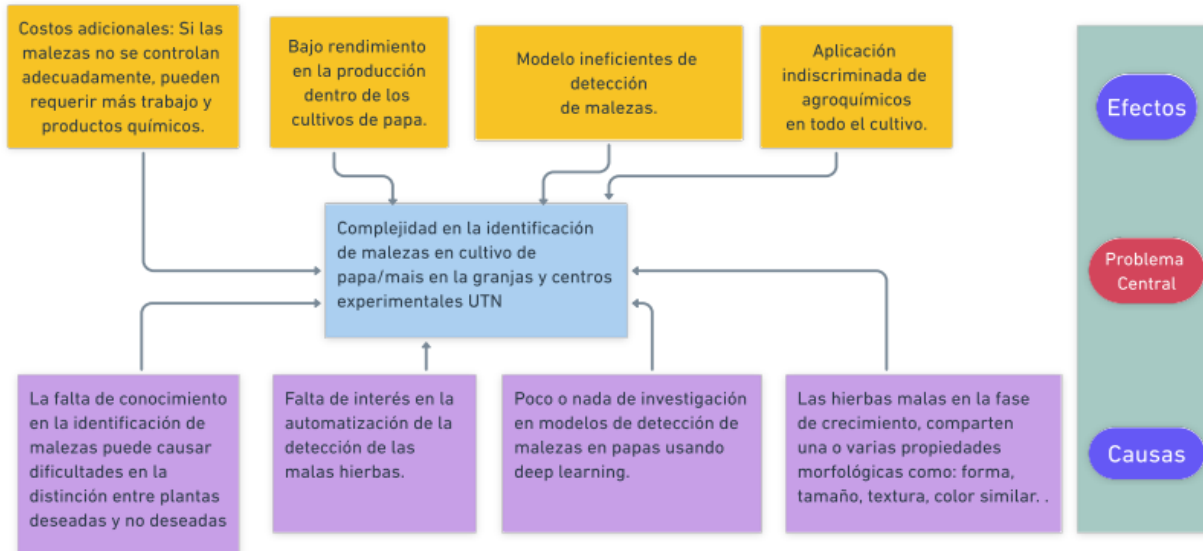
Para aplicar agroquímicos se utilizan equipos de maquinaria pesada, como tractores, avionetas o helicópteros. Los tractores pueden causar la erosión del suelo debido a la compactación (Basso & Pignaton de Freitas, 2020), las avionetas pueden alejar la aplicación del objetivo deseado y afectar a plantas y animales cercanos. Además, algunas malezas pueden volverse resistentes a los herbicidas comunes, lo que requiere el uso de compuestos más costosos y potentes (Hawkins et al., 2019).

Los desafíos en la detección y clasificación automática de cultivos y malezas se deben a que, en la fase de crecimiento, los cultivos y las malezas tienen propiedades morfológicas similares, como forma, tamaño, textura y color verde. Esto hace que sea difícil discriminar entre ellos. Además, hay otros factores, como la cromaticidad, el color y la textura que pueden verse afectados por la humedad, enfermedades y madurez de la hoja (Cervantes et al., 2017). También hay condiciones variables en el campo, como la iluminación, la sombra natural, la oclusión, las escalas, el entrecruzamiento y la superposición de las plantas. (I. García-Santillán et al., 2016), lo que complica aún más el reconocimiento de las plantas haciendo que la precisión de la de la clasificación se reduzca.

Para evidenciar de manera visual la clasificación e identificación de problemas ver la Figura 1.

Figura 1

Árbol de problemas



Objetivos

Objetivo General

Detectar malezas en cultivos de maíz y/o papa mediante el análisis de imágenes adquiridas por un dron basado en la arquitectura de red neuronal convolucional VGG.

Objetivos Específicos

- Elaborar un marco teórico que sustente las técnicas, herramientas y frameworks utilizadas para la detección automática de malezas.
- Desarrollar un modelo de una red neuronal convolucional (CNN) basado en la arquitectura VGG que permita la detección de malezas en cultivos de maíz y/o papa.
- Validar los resultados mediante métricas cuantitativas de rendimiento.

Alcance

Para comprender las herramientas utilizadas para la detección automática de malezas utilizando la arquitectura de red neuronal convolucional (CNN VGG) y conocer la situación actual en el campo del aprendizaje profundo y la agricultura de precisión, se elaborará una base teórica sobre los temas tratados en este proyecto. Este trabajo también incluye el uso de un dron Mavic Pro-2 equipado con una cámara para capturar imágenes georreferenciadas del cultivo en formato RGB de las diferentes clases de malezas que puedan existir en los cultivos de maíz y/o papa, según sus características morfológicas y de color.

Con la ayuda del dron se van a capturar mínimo 400 imágenes 4k de los cultivos de maíz y/o papa de los centros experimentales de la Universidad Técnica del Norte en donde exista gran presencia de malezas, lo cual servirán para el experimento de este proyecto, paralelo a esto, se identificará la altura y el clima con la que es más optima capturar las imágenes con el dron. Una vez obtenido el conjunto de imágenes se trabajará con la etiquetación de imágenes que es esencial para el aprendizaje automático profundo, ya que proporciona información sobre el contenido de las imágenes y permite a los algoritmos de Deep Learning aprender a reconocer y clasificar objetos y escenas. La complejidad de la etiquetación de imágenes varía dependiendo del contexto y del tipo de imágenes, pero en general puede ser un proceso tedioso y laborioso que requiere un gran esfuerzo y revisión humana para asegurar la precisión de las etiquetas.

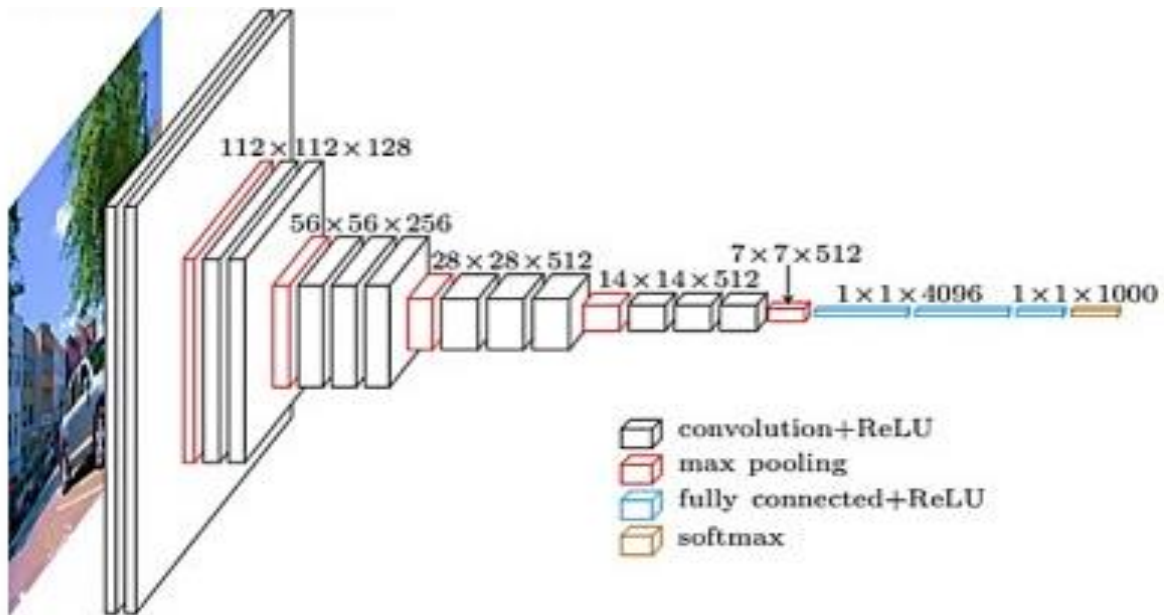
Las herramientas utilizadas para entrenar el modelo de Deep Learning son: Google Colaboratory y Python. Este lenguaje de programación cuenta con Keras y Tensor Flow, la cuales son librerías que facilitan la programación de modelos que usen redes neuronales.

Para el desarrollo del modelo, se tomará como base la CNN VGG, que es un tipo de red neuronal que se utiliza a menudo para la clasificación de imágenes y otros problemas de aprendizaje automático. La red neuronal VGG es una arquitectura de red neuronal convolucional desarrollada por Visual Geometry Group. Esta arquitectura ha sido ampliamente utilizada en la investigación y ha demostrado tener buen rendimiento en varias tareas de clasificación de imágenes (Bosch Rue et al., 2019).

En términos generales VGG16 comprende la entrada de datos, extracción de características (convoluciones) y finalmente la clasificación como se puede observar en la Figura 2.

Figura 2

Arquitectura de red neuronal



Nota: Adaptado de UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation, de Pravitasari Anindya, 2020. *Telkomnika* (<http://doi.org/10.12928/telkomnika.v18i3.14753>)(Pravitasari et al., 2020)

Metodología

El primer objetivo se logrará a través de una revisión de literatura basada en fuentes bibliográficas, que incluye descripciones de técnicas de aprendizaje profundo utilizadas en el proceso de clasificación e identificación de malezas en cultivo de papa, además de la clasificación según morfología, color y etapa de crecimiento. La documentación primaria incluye: manuales técnicos, trabajos de investigación, tesis y normativa nacional, por lo que las referencias contribuirán a la calidad del trabajo presentado.

Para cumplir el segundo objetivo, recolección de datos, se utilizará la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), a fin de verificar la calidad de los datos, y posteriormente procesar las imágenes con la finalidad de usarlas en una arquitectura de alto rendimiento en donde se usara ciertas técnicas de deep Learning para la clasificación y detección de patrones en imágenes relacionadas a malezas, ajustando sus parámetros con el propósito de mejorar los resultados.

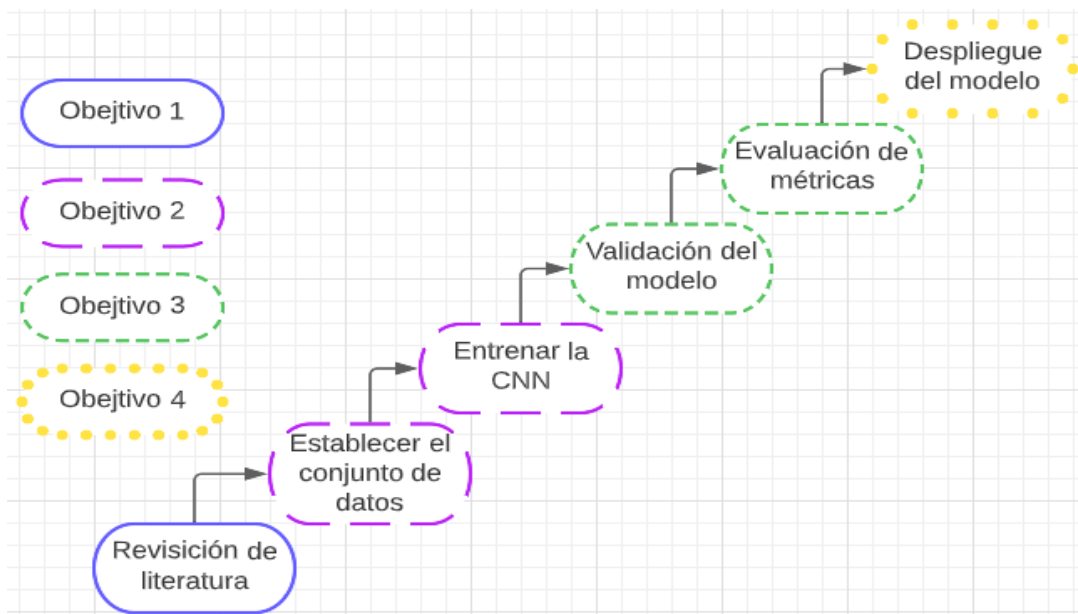
Por otro lado, para cumplir, el tercer objetivo se realizará un análisis estadístico de las métricas obtenidas en la fase de pruebas del modelo, como, por ejemplo, la precisión y pérdida junto con su visualización a través de gráficas y la matriz de confusión, confirmando un buen rendimiento, precisión y errores mínimos.

Por último, para cumplir el cuarto objetivo el modelo obtenido se desplegará en una aplicación web, a fin de verificar el funcionamiento del modelo.

Para el desarrollo del proyecto se va a seguir la siguiente metodología que se resumen a continuación en la Error! Reference source not found..

Figura 3

Síntesis del desarrollo del proyecto



Justificación

El objetivo 9 de la Agenda 2030 para el Desarrollo Sostenible tiene como propósito promover el crecimiento económico sostenible, inclusivo y sostenible, así como, el empleo pleno y productivo y el trabajo decente para todos. La Sección 9. enfatiza en particular la modernización de la infraestructura y la transformación en industrias

sostenibles, el uso más eficiente de los recursos y la promoción de la adopción de tecnología.

El Plan Estratégico de Desarrollo y Planificación Regional de Imbabura tiene como objetivo "incrementar el desarrollo productivo de la agricultura, la artesanía, la industria, el sector de servicios y el turismo en la provincia de Imbabura". Según el III Censo Agropecuario de 2001, la provincia de Imbabura cuenta con 33.786 Unidades de Producción Agropecuaria (UPA), de las cuales tienen una superficie de 283.659 hectáreas (Prefectura de Imbabura, 2014), lo que indica su importancia para reconocer la modernización de la producción agrícola y los medios técnicos no implementados.

Para ello, este proyecto tiene como objetivo brindar y apoyar al sector agropecuario con la detección automática de malezas utilizando métodos de aprendizaje profundo aplicados a la finca "La Pradera" en Chaltura (Imbabura), a cargo de la Facultad de Ciencias Agropecuarias y Ambientales (FICAYA). lo que también favorece a sus estudiantes y docentes con la eficiencia y calidad del producto y a su vez promueve la innovación tecnológica agropecuaria en la provincia y el país.

Justificación Tecnológica

Con el desarrollo de este trabajo se busca contribuir a la modernización y optimización de procesos agrícolas utilizando tecnologías como inteligencia artificial y Deep Learning.

Justificación Teórica

Las técnicas basadas en el aprendizaje profundo están reemplazando las técnicas tradicionales de aprendizaje automático para detectar malezas en tiempo real con el desarrollo de nuevos modelos y el aumento de la potencia computacional. Están surgiendo más modelos híbridos de aprendizaje automático, que utilizan los beneficios de diferentes técnicas. Más conjuntos de

datos de imágenes de cultivos y malezas a gran escala están disponibles en línea ahora, y esto proporciona más datos y oportunidades para que los investigadores e ingenieros se unan y contribuyan a este campo.(B. Liu & Bruch, 2020)

Justificación Metodológica

El presente proyecto se lo va a realizar siguiendo la metodología CRISP-DM(Cross-Industry Standard Process for Data Mining) que incluye descripciones de las fases a seguir, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas, por otro lado CRISP-DM permite crear un modelo de minería de datos que se adapte a necesidades concretas de los problemas a tratar (IMB Corporation, 2021).

CAPÍTULO I

Marco Teórico

1.1 Fundamentación Teórica

1.1.1 Malezas

En el ámbito agrícola, las malezas se definen como plantas no deseadas que crecen en lugares donde no deberían. Según la literatura científica, estas plantas pueden generar pérdidas de rendimiento en los cultivos si no se manejan adecuadamente. Por lo tanto, es crucial comprender su biología y ecología para poder aplicar estrategias de control efectivas (Schwartz-Lazaro et al., 2021).

Las malezas pueden competir con los cultivos por los nutrientes y recursos del suelo, lo que puede disminuir significativamente la calidad y cantidad de la producción agrícola (Mahmudul Hasan et al., 2022). Además, el control manual de malezas puede ser muy costoso y laborioso en áreas de cultivo extensas, lo que ha llevado a la necesidad de desarrollar sistemas automáticos de gestión de malezas. La detección de malezas en imágenes puede ser un desafío debido a la similitud en el color, la textura y la forma entre las malezas y los cultivos, así como a las variaciones en las condiciones de imagen. Sin embargo, con el uso de técnicas avanzadas de aprendizaje automático, es posible lograr una alta precisión en la identificación de malezas y la eliminación selectiva de las mismas, mejorando así la eficiencia y la rentabilidad de la producción agrícola (Mahmudul Hasan et al., 2022).

La identificación de las malas hierbas es la clave de un programa eficaz de control de las malas hierbas del maíz. La identificación errónea de una mala hierba puede suponer la diferencia entre beneficios y pérdidas. Aunque el ciclo de vida de una mala hierba, incluido su modo de reproducción, es la característica de identificación más importante, a veces es necesario conocer la especie exacta antes de elegir la medida de gestión de malezas.

1.1.2 Control de malezas

- a) Los **daños** que puede ocasionar un inadecuado control de malezas pueden ser: refugio de plagas, competir por agua luz y espacio, reduce la calidad de la cosecha, incremento los costos de producción y requiere mayor control sanitario
- b) El **control manual** de malezas puede ser efectivo en pequeñas áreas, pero en grandes extensiones de tierra se hace necesario el uso de herbicidas y/o la implementación de técnicas de labranza. Sin embargo, el uso de herbicidas debe hacerse con precaución para evitar dañar el cultivo y el medio ambiente. Por lo tanto, es importante implementar un plan de manejo integrado de malezas que combine diferentes métodos de control para lograr una gestión eficiente y sostenible de las malezas en los cultivos de papa (Rodríguez, 2020).
- c) El **control cultural** se da cuando se elige desde un principio semillas de calidad; preparar adecuadamente el suelo; con respecto a la siembra debe ser oportuna en periodo de lluvias; con las distancias de siembra debe existir de 25 a 40 cm entre plantas por 90 a 120 cm entre surcos; para la fertilización se debe tomar en cuenta las necesidades del cultivo, paralelo a esto hay que realizar labores culturales oportunas, sin descuidar la importancia de la rotación de cultivos que ayudan a interrumpir los ciclos vegetativos de las malezas (INIAP, 2018).
- d) **Control mecánico** consiste en la utilización de herramientas manuales, tracción animal y maquinaria agrícola, algunas labores que contribuyen a erradicar las malezas son el rascadillo, medio aporque y aporque, actividades que deben ser realizadas oportunamente (INIAP, 2014).
- e) El **Control automático** de las malas hierbas emplea la inteligencia artificial y la visión por ordenador, lo que requiere el empleo de un subsistema de sensores capaz de reconocer y discriminar las plantas agrícolas de las

malas hierbas. Esta estrategia es ventajosa para reducir la contaminación ambiental y lograr un desarrollo agrícola sostenible (S. Liu et al., 2022; Su, 2020).

1.1.3 Tipos de malezas en cultivo de maíz

(Cornell CALS, 2021) menciona que existen diferentes tipos de malezas que pueden afectar los cultivos de maíz. Estas malezas pueden ser anuales, perennes, bienales o simples. Algunas malezas anuales, como las quenopodiáceas o los rabos de zorra, pueden ser controladas con herbicidas y una o dos labores de cultivo. Sin embargo, otras malezas perennes, como la leche común y la grama, pueden ser difíciles de controlar. Las malezas bienales, como la zanahoria silvestre, no persisten en campos que son arados, pero pueden ser un problema en campos con ausencia de labores de cultivo. Por lo tanto, es importante conocer los diferentes tipos de malezas y aplicar las técnicas de control adecuadas para minimizar su impacto.

Según (Lingenfelter, 2022) el pokeweed (hierba carmín) es una maleza difícil de controlar debido a su gran tamaño y sistema de raíces profundas y extensas. También menciona que el pokeweed puede reducir el rendimiento de los cultivos de maíz, además de ser un problema en el control de malezas debido a su capacidad para producir una gran cantidad de semillas.

Algunas de las malezas más comunes en los cultivos de maíz se pueden ver en la Tabla 1.

Tabla 1

Malezas en cultivos de maíz.

Nombre científico	Nombre común
<i>Ceratophyllum demersum</i>	cola de zorro

<i>Asclepias curassavica</i>	algodoncillo
<i>Daucus carota</i>	zanahoria silvestre
<i>Phytolacca americana</i>	hierba carmín

1.1.4 Tipos de malezas en cultivos de papa

El manejo eficaz de plantas no deseadas en cultivos de patata es de suma importancia. Estas plantas compiten directamente con las patatas por recursos esenciales como el agua, los nutrientes del suelo y la luz solar. Además, pueden servir como refugio para insectos perjudiciales y actuar como vectores de enfermedades que afectan a las plantas de patata.. Según (Government of New Brunswick, 2019), entre las especies de malezas más comunes en campos de papa se encuentran la belladona, la mostaza silvestre, la hierba de los corderos, la hierba cochinilla y la cola de zorro. Para implementar un programa efectivo de control de malezas, se deben considerar las especies problemáticas presentes en el campo, evitando en lo posible el cultivo de papa en campos que contengan malezas perennes.

El control de malezas en los cultivos de papa es fundamental para garantizar una buena producción y evitar pérdidas económicas. Según (Rodríguez, 2020), las malezas en los cultivos de papa pueden competir con el cultivo por los recursos del suelo, reducir la calidad de la cosecha y aumentar los costos de producción. Además, las malezas pueden actuar como hospederos de plagas y enfermedades, lo que puede afectar aún más la salud del cultivo.

Las malezas más comunes existentes en los cultivos de papa se pueden ver en la Tabla 2.

Tabla 2*Malezas en cultivo de papa*

Nombre Científico	Nombre común
<i>Atropa belladonna</i>	belladona
<i>Sinapis alba</i>	mostaza silvestre
<i>Amaranthus caudatus</i>	amaranto
<i>Ambrosia ambrosioides</i>	ambrosia
<i>Ceratophyllum demersum</i>	cola de zorro

1.1.5 Imágenes por dron

Las imágenes aéreas tomadas por drones, también conocidas como fotografías aéreas tomadas por drones, son capturas visuales tomadas desde vehículos aéreos no tripulados. Estas fotografías brindan una visión distintiva y detallada del terreno o área fotografiada, lo que es útil para aplicaciones como la cartografía, la agricultura de precisión, la inspección de infraestructuras y la vigilancia ambiental, (Sánchez & Calderón, 2020).

Las imágenes multiespectrales capturadas por drones son cruciales para una variedad de aplicaciones, incluida la clasificación de terrenos y la vigilancia ambiental. La tecnología, que ha permitido una captura eficiente de datos para análisis detallados, ha evolucionado desde los globos y aviones tripulados originales hasta los drones actuales. Los algoritmos de clasificación supervisados o no supervisados se utilizan para extraer información útil de estas imágenes, (Oliveros, 2022).

1.1.6 Vehículos aéreos no tripulados (Drones)

Un tipo de vehículo aéreo que ha ganado importancia en la actualidad según (Harvard University, 2015) es el Vehículo Aéreo No Tripulado (UAV), también conocido como drone. A diferencia de los aviones tripulados, el UAV no necesita de un piloto humano a bordo y en su lugar, utiliza fuerzas aerodinámicas para mantenerse en el aire. El UAV puede volar de forma autónoma o ser controlado a distancia, y puede ser diseñado para ser de un solo uso o recuperable. Además, es capaz de transportar una variedad de cargas útiles, tanto letales como no letales.

(Lutkevich, 2021) acota que los sistemas de aeronaves no tripuladas, conocidos comúnmente como drones, son esencialmente robots voladores que pueden ser operados de forma remota o volar de manera autónoma a través de planes de vuelo preestablecidos mediante software integrado en sus sistemas, que trabajan en conjunto con sensores a bordo y un sistema de posicionamiento global (GPS).

Por otro lado, el (New England Institute of Technology, 2022) señala que los *Unmanned Aerial Vehicles* (UAV), se han utilizado cada vez más para una variedad de propósitos, como la entrega de productos, la vigilancia, la fotografía y filmación aérea, la agricultura, las carreras de drones, exploraciones geográficas, el mapeo de incendios forestales y otras aplicaciones.

1.2 Redes Neuronales Convolucionales

1.2.1 Inteligencia Artificial (IA)

Según(Jiang et al., 2022), durante más de seis décadas, se ha llevado a cabo una extensa investigación en el campo de la inteligencia artificial, logrando avances significativos tanto en términos teóricos como en aplicaciones prácticas. La IA se ha integrado en una amplia gama de áreas y se reconoce como una habilidad fundamental para el futuro. Según la fuente de (Artificial Intellenge (AI) Market, 2023), se proyecta que el mercado de la inteligencia artificial alcance los

190.000 millones de dólares para el año 2025, con una tasa de crecimiento anual compuesta superior al 36% entre 2018 y 2025.

Aunque existen diversas interpretaciones sobre la inteligencia artificial, hay un amplio consenso en considerar que su núcleo abarca teorías, métodos, tecnologías y aplicaciones de investigación destinadas a simular, ampliar y mejorar la inteligencia humana. En la actualidad, el impacto de la IA en la vida humana es cada vez más profundo. Al igual que los motores de vapor en la era del vapor, los generadores en la era de la electricidad y las computadoras en la era de la información, la inteligencia artificial se ha convertido en un pilar tecnológico en la era contemporánea y más allá (Jiang et al., 2022).

1.2.2 Vision Artificial

El papel clave que desempeña la visión por computadora en la rápida transformación digital de la industria agrícola. También menciona que esta tecnología, especialmente el uso de enfoques de aprendizaje profundo permite realizar una variedad de actividades agrícolas de manera automática y precisa. La tecnología de visión computarizada ha emergido como un recurso fundamental en el sector agrícola, abarcando diversas aplicaciones desde la evaluación de la calidad de semillas hasta la predicción de cosechas. Esta herramienta impulsa significativamente la productividad y eficiencia en la agricultura. No obstante, es crucial enfatizar que para obtener resultados óptimos, los modelos deben basarse en datos de alta calidad y ofrecer soluciones inmediatas, aspectos clave para el éxito de la implementación de sistemas de visión por computadora en el ámbito agrícola, (Matt Russell, 2018; Moncayo & Karla, 2021).

1.2.3 Redes neuronales convolucionales (CNN)

Las redes convolucionales Convolutional Neural Networks (CNN) son capaces de procesar datos secuenciales, como la secuencialidad temporal de las muestras de voz y la secuencialidad espacial de los píxeles de las

imágenes. La información relevante se puede encontrar en diferentes posiciones de los valores de cada muestra, lo que permite a las CNN detectar características significativas cambiadas de escala, como los ojos de los animales que pueden estar en diferentes posiciones de la imagen. Las CNN realizan la convolución mediante el uso de un filtro o kernel, que se desplaza por los valores de las muestras. El filtro es bidimensional en el caso de las imágenes y se desplaza por las dos dimensiones de la imagen. Es importante tener en cuenta que el diseño de las CNN se basa en la idea de que la información relevante se encuentra en diferentes posiciones de los datos y que las características pueden variar en escala (Bobadilla, 2020).

De acuerdo con (F.-F. Li et al., 2021) Las redes neuronales convolucionales, también conocidas como ConvNets, son un tipo especializado de arquitectura de redes neuronales. Estas se han diseñado específicamente para analizar y procesar información que presenta patrones secuenciales o estructurados, como imágenes o señales de audio. Aunque tienen muchas similitudes con las redes neuronales tradicionales, también presentan algunas diferencias clave en su diseño y funcionamiento.

En lugar de estar formadas por neuronas simples, las ConvNets están compuestas por capas que transforman el volumen de la imagen en un volumen de salida. Estas capas utilizan una función diferenciable para transformar un volumen de entrada 3D en un volumen de salida 3D. Algunas de las capas más populares son CONV, FC, RELU y POOL, que pueden tener o no parámetros y hiperparámetros adicionales. Es importante tener en cuenta que los trucos y consejos desarrollados para las redes neuronales tradicionales también son aplicables a las ConvNets, lo que permite mejorar su rendimiento y precisión en la tarea de clasificación de imágenes (F.-F. Li et al., 2021).

Según (Xin & Wang, 2019), los pasos para entrenar un modelo de red neuronal convolucional son los siguientes:

Entrada: El conjunto de entrenamiento es una colección de N imágenes, cada una con una etiqueta de imagen que pertenece a una de las K etiquetas de clasificación.

Aprendizaje: El aprendizaje de un modelo se refiere al uso de un conjunto de entrenamiento para aprender cómo es cada clase. Este proceso se conoce como clasificador de entrenamiento o aprendizaje de un modelo.

Evaluación: Un clasificador se emplea para prever las categorías de imágenes desconocidas y para valorar la calidad de estos. Las categorías previstas por el clasificador son cotejadas con las categorías verdaderas de la imagen.

1.2.4 Componentes, arquitectura y técnicas de entrenamiento

(F.-F. Li et al., 2021) describe la arquitectura de una red neuronal convolucional (ConvNet) simple como [INPUT - CONV - RELU - POOL - FC]. A continuación, se detallan los componentes de la arquitectura VGG16 que detallas (Keras, 2021) en su página oficial:

- ***include_top***: determina si se incluyen las 3 capas completamente conectadas en la parte superior de la red.
- ***weights***: opciones de inicialización aleatoria, pre-entrenamiento en ImageNet, o la ruta al archivo de pesos a cargar.
- ***input_tensor***: tensor Keras opcional para utilizar como entrada de imagen para el modelo.
- ***input_shape***: tupla de forma opcional para especificar la forma de entrada si *include_top* es False.
- ***pooling***: modo de agrupación opcional para la extracción de características cuando *include_top* es False.

- **classes:** número opcional de clases en las que clasificar las imágenes si `include_top` es *true*.
- **classifier_activation:** función de activación opcional para la capa "superior" si `include_top` es *true*.
- **Transfer Learning**

El aprendizaje por transferencia técnica que trata sobre tomar características aprendidas en un problema y utilizarlas en un problema nuevo y similar. Se suele realizarse para tareas en las que el conjunto de datos es demasiado escaso para entrenar un modelo completo desde cero. (keras, 2020)

La transferencia de aprendizaje ha demostrado ser efectiva en la mejora del rendimiento de los modelos de reconocimiento de patrones en imágenes. Según los hallazgos de (Bazan Yaranga, 2018), el uso de redes pre-entrenadas para la transferencia de aprendizaje puede mejorar significativamente el rendimiento del modelo. Además, esta técnica puede ser valiosa en otros problemas de reconocimiento de patrones en imágenes ya que las redes pre-entrenadas pueden contener características generales útiles para la tarea de interés y pueden reducir el tiempo y recursos necesarios para entrenar un modelo desde cero.

- **Regularización**

La regularización para evitar el sobreajuste. La regularización más común utilizada en la CNN es la L2, que penaliza los pesos más grandes en la red.

(Kukačka et al., 2017), menciona que el objetivo de la regularización es evitar el sobreajuste de los modelos de aprendizaje profundo, que ocurre cuando el modelo es demasiado complejo y se ajusta demasiado a los

datos de entrenamiento, lo que puede llevar a un rendimiento deficiente en los datos de prueba o en nuevos datos.

Lo que consigue la regularización es penalizar las predicciones que realiza nuestra red durante el entrenamiento para que no piense que el training set es la verdad absoluta del estudio y, además, sepa generalizar mejor con otros datasets futuros (KeepCoding, 2021).

- **Regularización L1 y L2:** (Shewchuk, 2020) menciona que son técnicas utilizadas para prevenir el sobreajuste en los modelos. La regularización L1 penaliza los coeficientes de regresión que no son importantes y los reduce a cero, lo que lleva a un modelo más simple y una selección de características más rigurosa. Por otro lado, la regularización L2 reduce los coeficientes de regresión sin eliminarlos por completo, lo que puede ser útil para mantener la importancia de ciertas características. Ambas técnicas mejoran la generalización del modelo al evitar que exista un overfitting a los datos que se usa entrenar el modelo.

- **Métodos que afectan los términos de regularización**

Las técnicas de regularización para redes neuronales profundas mencionadas por (Yam, 2022) son:

Regularización de rango bajo: Reducir la complejidad de las redes neuronales mediante descomposición en valores singulares, para evitar el sobreajuste.

Suavidad temporal: Penalizar cambios abruptos en las valoraciones de las acciones a lo largo del tiempo, para promover valoraciones más consistentes y estables. Esto previene que la red oscile fácilmente entre valoraciones dispares de la misma acción en pasos de tiempo consecutivos.

- **Funciones de activación ReLU:**

La función de activación ReLU (Rectified Linear Unit) en todas sus capas convolucionales, excepto en la capa final. (Grimstad & Andersson, 2019) señalan que el ajuste de límites puede reducir considerablemente los tiempos de solución, y que las redes ReLU de pequeño tamaño son adecuadas como modelos sustitutos en programas lineales de enteros mixtos.

- **Max Pooling:**

La técnica de reducción de dimensionalidad conocida como Max Pooling 2D es utilizada para reducir las características a la mitad. Los parámetros que se pueden ajustar en esta capa incluyen los tamaños de ventana de pooling (pool_size), el stride (stride) y el tipo de relleno (padding), (keras, 2021).

- **Ajuste de aprendizaje:**

Las CNN ajustan la tasa de aprendizaje (**Learning rate**) durante el entrenamiento para mejorar el rendimiento.

La tasa de aprendizaje es crucial en los sistemas de aprendizaje en línea. Un ritmo de aprendizaje alto permite adaptarse rápidamente a los nuevos datos, pero también puede hacer que se olviden los antiguos y se sea más sensible al ruido. Una tasa de aprendizaje baja puede evitar el ruido, pero también olvida rápidamente los datos antiguos. Una tasa de aprendizaje demasiado alta puede hacer que el algoritmo diverja y no encuentre una buena solución (Géron, 2017).

1.2.5 Aprendizaje supervisado y no supervisado

Según (IBM, 2023), el aprendizaje supervisado y aprendizaje no supervisado son dos enfoques fundamentales en el campo de la inteligencia artificial y el aprendizaje automático.

- **Aprendizaje Supervisado:**

El modelo es entrenado con un dataset etiquetado, donde cada ejemplo tiene una etiqueta o resultado conocido. El objetivo es aprender una función que mapee las entradas a las salidas correctas.

Ejemplos comunes de algoritmos de aprendizaje supervisado incluyen *regresión lineal*, *máquinas de soporte vectorial (SVM)* y *redes neuronales*.

- **Aprendizaje No Supervisado:**

En este caso, el modelo se entrega con datos no etiquetados, sin información sobre las salidas correctas. El fin es encontrar patrones o agrupaciones en los datos.

Ejemplos de algoritmos de aprendizaje no supervisado incluyen *clustering (agrupamiento)*, *reducción de dimensionalidad* y *análisis de componentes principales (PCA)*.

1.2.6 Herramientas y bibliotecas para trabajar con redes neuronales convolucionales

- **Tensorflow**

Según (Google Cloud, 2022), TensorFlow es una biblioteca de código abierto de extremo a extremo para el aprendizaje automático. Por otro lado (edX, 2022) menciona que Tensorflow fue creada por Google Brain y que en principio fue concebida para su uso interno en Google, pero luego fue publicada bajo la licencia de código abierto Apache 2.0. La biblioteca usa redes neuronales para el aprendizaje profundo y puede ser utilizada para una variedad de aplicaciones, por ejemplo, la búsqueda de algoritmos, el análisis, procesamiento y clasificación de imágenes, y muchas otras.

TensorFlow facilita la creación de modelos de aprendizaje automático para computadoras de escritorio, dispositivos móviles, la web y la nube, sin importar si se es principiante o experto (TensorFlow, 2023).

PyTorch es una biblioteca de aprendizaje profundo de código abierto desarrollada por Facebook's AI Research, que se basa en el lenguaje Python (PyTorch, 2021). Este marco de trabajo de software proporciona a los desarrolladores una amplia gama de herramientas y funciones destinadas a facilitar la creación y el entrenamiento de modelos de detección de objetos (PyTorch, 2021). Utilizando PyTorch, los investigadores y los desarrolladores pueden implementar de manera eficiente redes neuronales convolucionales (CNNs, por sus siglas en inglés), que son ampliamente utilizadas en la detección precisa de objetos en imágenes y videos (PyTorch, 2021). Además, este framework ofrece características que permiten el procesamiento de datos, la manipulación de imágenes, el cálculo de alto rendimiento utilizando GPU y la visualización de resultados, lo que contribuye a la construcción efectiva de sistemas de detección de objetos (PyTorch, 2021).

- **Keras**

Keras como una API de aprendizaje profundo de alto nivel ha ganado popularidad debido a su sintaxis simple y facilidad de uso para construir modelos de aprendizaje profundo. Normalmente, Keras se utiliza en combinación con una biblioteca de bajo nivel como TensorFlow. Para usar Keras, primero es necesario instalar TensorFlow, y al importar Keras, se mostrará el backend que se utilizó para instalar la biblioteca (IBM, 2023b).

Keras es una API de alto nivel de TensorFlow para construir y entrenar modelos de aprendizaje profundo (TensorFlow, 2022). La rapidez para obtener resultados es fundamental en la investigación y (Torres, 2020) Keras proporciona una API fácil de usar (Allaire & Chollet, 2021) que permite desarrollar prototipos de modelos de manera ágil. Además, su

versatilidad permite ejecutar el código tanto en la CPU o en la vram sin restricciones. Una de las características principales de Keras es su soporte integrado para redes convolucionales y recurrentes, lo que la hace ideal para aplicaciones de visión por computador y procesamiento de secuencias. Asimismo, Keras admite arquitecturas de red arbitrarias, como modelos multientrada o multisalida, capas compartidas, modelos compartidos, entre otras opciones (Allaire & Chollet, 2021).

Para (Unipython, 2021), la creación de modelos de aprendizaje profundo en Keras se puede resumir de la siguiente manera:

- (1) **Definir el modelo:** se debe crear un modelo secuencial y agregar las capas necesarias que conformen la arquitectura deseada de la red neuronal.
- (2) **Compilar el modelo:** después de definir la arquitectura del modelo, se debe compilar para especificar la función de pérdida y los optimizadores adecuados. Para ello, se llama a la función *compile()* en el modelo.
- (3) **Entrenar el modelo:** una vez que se ha compilado el modelo, se puede ajustar mediante el entrenamiento con una muestra de datos. Esto se logra llamando a la función *fit()* en el modelo.
- (4) **Generar predicciones:** por último, una vez entrenado el modelo, se lo utiliza para realizar predicciones en datos no vistos con anterioridad. Esto se logra llamando a la función *evaluate()* o *predict()* en el modelo.

1.2.7 Red Neuronal Convolucional VGG

El modelo VGG son las siglas del Visual Geometry Group de Oxford (Matt Russell, 2018). Es una topología de redes neuronales convolucionales que es usado en la clasificación de imágenes. Fue presentada en 2014 por Simonyan y

Zisserman y se caracteriza por tener una profundidad muy elevada en comparación con otras arquitecturas similares en su época. La red VGG se destacó por su simplicidad y eficacia, ya que se utilizaron filtros de 3x3 y capas de pooling de 2x2 que posee stride 2 para disminuir el tamaño de los mapas de características. VGG se considera una de las redes más influyentes y ha servido como base para muchas otras arquitecturas, (VISO AI, 2021).

De acuerdo con (Bosch Rue et al., 2019), una de las características más notables de esta red neuronal es que, en lugar de tener muchos hiperparámetros, sus autores utilizaron una red muy simple compuesta únicamente por capas convolucionales con filtros de 3x3 y stride $s = 1$, y capas de pooling con función de agrupamiento max-pooling de 2x2 y stride $s = 2$. A pesar de su complejidad, VGG es atractiva debido a su simplicidad. VGG es bastante uniforme, con capas convolucionales precedidas por capas de pooling que comprimen la dimensión de los volúmenes. En total, la red VGG tiene 138 millones de parámetros.

En cada capa convolucional de la arquitectura, el número de filtros utilizados se duplica con respecto a la capa anterior. Los valores comienzan en 64 y aumentan a 128, 256 y finalmente 512. Esto construye un principio sencillo en la forma de la topología de la red. Una de las desventajas puntuales de la red VGG es el extenso número de parámetros que posee y se debe buscar alguna manera de ajustar durante el entrenamiento. En la literatura, hay referencias a las redes VGG-16 y VGG-19. VGG-19 es una versión aún más profunda que VGG-16, pero la precisión obtenida por ambas es muy similar y generalmente se utiliza la red VGG-16 (Bosch Rue et al., 2019).

Conocer los detalles sobre la arquitectura VGG en Deep Learning es de suma importancia, puesto que esta es una de las arquitecturas más utilizadas para el manejo de los macrodatos en el aprendizaje profundo de redes neuronales convolucionales (Villalba, 2020).

1.2.8 Características de la arquitectura VGG

- **Descenso de gradiente estocástico (SGD):** El algoritmo de descenso de gradiente estocástico (SGD) se utiliza para encontrar los valores de los parámetros que minimizan una función de costo en aprendizaje automático y redes neuronales (Bushaev, 2017). Durante el entrenamiento, los modelos aprenden de los datos proporcionados y la función de costo, que mide la precisión del modelo en cada iteración, es esencial para el proceso de optimización. El descenso de gradiente utiliza esta función de costo para ajustar los parámetros del modelo de manera iterativa, mejorando su precisión con el tiempo, (IMB, 2024).
- **Momentum:** El SGD a menudo sufre de fluctuaciones en la dirección de la gradiente, lo que ralentiza su convergencia. Para mejorar el rendimiento del SGD, se puede introducir un término de "momentum" en la actualización de los parámetros. Este término tiene en cuenta la dirección y magnitud de la actualización anterior, permitiendo suavizar las fluctuaciones y acelerar la convergencia del algoritmo (Bushaev, 2017).
- **Aplicaciones y resultados:** La arquitectura VGG ha sido utilizada con éxito en una gran variedad de aplicaciones de aprendizaje automático, que abarca la clasificación de imágenes.
 - a) El estudio de (Khan et al., 2020) se enfoca en un método para detectar y clasificar enfermedades de las hojas del pepino, que consta de cinco etapas: mejora de la imagen, segmentación de las manchas infectadas, extracción y selección de características, y clasificación. Se utiliza VGG y se seleccionan características destacadas. El método logró una precisión de clasificación del 98,08% en 10,52 segundos.
 - b) El cáncer de piel es una enfermedad peligrosa y se puede curar si se detecta a tiempo. Para detectar lesiones cutáneas de manera automática, se ha utilizado la visualización con cribado

clínico por análisis dermatoscópico. En este proceso se han utilizado diferentes métodos, como un clasificador de aprendizaje profundo con la red neuronal convolucional VGG-16. Para aumentar los datos, se añadieron pesos y se redimensionó la imagen. Este clasificador ha demostrado ser efectivo en la detección temprana de melanomas benignos y malignos (Manasa & Murthy, 2021).

c) El diagnóstico temprano de tumores cerebrales según (Majib et al., 2021) es crucial para la supervivencia de los pacientes, pero la segmentación y el diagnóstico de las áreas tumorales a partir de imágenes de resonancia magnética es un proceso tedioso que depende de la pericia del especialista. Por ello, se han desarrollado tecnologías asistidas por ordenador para mejorar la precisión del diagnóstico. Se analizaron diferentes modelos de aprendizaje de transferencia y se propuso un clasificador apilado llamado VGG-SCNet, basado en la red neuronal VGG-16, que obtuvo altas puntuaciones de precisión, recuperación y f1 en la clasificación de tumores cerebrales sin intervención humana.

- **Comparación con otras arquitecturas:**

a) Para (Moscosio, 2020) AlexNet, VGG y NiN son tres arquitecturas de redes neuronales profundas. La principal diferencia entre ellas radica en la estructura de sus capas convolucionales. AlexNet utiliza capas convolucionales con tamaños de kernel grandes y saltos de tamaño 4, mientras que VGG utiliza capas convolucionales con tamaños de kernel pequeños y saltos de tamaño 1. Por otro lado, NiN utiliza capas convolucionales con tamaños de kernel 1x1 y 3x3, en lugar de utilizar capas totalmente conectadas. En términos generales, AlexNet y VGG se enfocan

en aumentar la profundidad de la red, mientras que NiN se enfoca en reducir la cantidad de parámetros y aumentar la eficiencia computacional.

- b) VGG-19, PlainNet y ResNet de 34 capas son arquitecturas utilizadas en tareas de visión por computadora. La principal diferencia entre ellas se encuentra en la cantidad de capas y la forma en que estas capas se organizan. VGG-19 y PlainNet se enfocan en la profundidad de la red, mientras que ResNet se enfoca en evitar la degradación del rendimiento a medida que se profundiza en la red mediante el uso de conexiones residuales (Moscosio, 2020).

1.3 Normas y Metodologías de Desarrollo

1.3.1 Proceso Interindustrial para minería de Datos (CRISP-DM)

Con el avance tecnológico, las organizaciones han podido almacenar grandes cantidades de datos. Sin embargo, surge el problema de cómo analizar estos conjuntos información para extraer información relevante para una adecuada toma de decisiones en situaciones reales. Para abordar este desafío, existen diferentes metodologías que ayudan a analizar los datos y extraer conocimiento, una de ellas es CRISP-DM (acrónimo de Cross Industry Standard Process for Data Mining) (Espinosa Zúñiga, 2020).

CRISP-DM es un marco de trabajo bien establecido para guiar los trabajos de minería de datos, este método constituye seis fases: comprensión; del negocio, de los datos; elaboración de los datos; modelado; evaluación y despliegue del modelo. La metodología CRISDM es ampliamente utilizada en el campo de la minería de datos para asegurar la eficacia y la eficiencia en el proceso de análisis de datos (IBM, 2021).

1.3.2 Fases de CRISP-DM

CRISP-DM es uno de los ciclos de vida más populares que se utilizan para el análisis predictivo(Oktavius et al., 2022).

Para (Amazon Web Services, 2023; Caio, 2022; HOTZ, 2023; Jiménez & Merino, 2023) las fases que conforma CRISP-DM son:

- 1) **Entendimiento del negocio:** Esta fase implica establecer claramente la meta del proyecto y determinar los requerimientos empresariales. Su propósito es lograr que todos los participantes del proyecto estén en sintonía y comprendan la dirección y las expectativas del mismo.
 - **Determinar los objetivos empresariales:** Comprender claramente los objetivos empresariales y lo que el cliente quiere conseguir desde una perspectiva empresarial, y definir los criterios de éxito en consecuencia.
 - **Evaluar la situación:** En esta etapa se evalúan los medios existentes, se determinan las exigencias del proyecto, se anticipan posibles obstáculos y sus soluciones, y se realiza una valoración económica que sopesa las ventajas frente a los gastos previstos.
 - **Determinar los objetivos de la minería de datos:** Se establecen los criterios de éxito tanto desde el enfoque técnico de la minería de datos como desde la perspectiva de las metas comerciales. Esto asegura una evaluación integral del proyecto, considerando tanto los aspectos técnicos como los objetivos del negocio.
 - **Elaborar el plan del proyecto:** Seleccionar tecnologías y herramientas y crear un plan detallado para cada fase del proyecto para garantizar el éxito de este.

2) **Entendimiento de los datos:** Se realiza una recolección de datos, se identifican las fuentes de datos y se establecen los formatos de los datos para evitar la pérdida de información relevante.

- **Reúna los datos iniciales:** Recopile la información requerida y, si es preciso, impórtela a las plataformas de análisis que utilizará. Este paso implica la adquisición de los datos relevantes y su preparación para el proceso analítico subsiguiente.
- **Describa los datos:** Realice una inspección inicial de la información obtenida. Registre las características básicas observadas, incluyendo el formato de los datos, la cantidad de entradas y la identificación de los diferentes campos o variables presentes en el conjunto de datos.
- **Explorar los datos:** Analizar detalladamente la información. Generar preguntas sobre los datos, crear representaciones visuales y descubra las conexiones existentes entre las distintas variables. Este proceso implica una exploración más profunda para comprender mejor la estructura y el contenido de los datos.
- **Validar la calidad de los datos:** ¿hasta qué punto están limpios o sucios?

3) **Preparación de los datos:** Los datos son organizados y preparados para ser analizados, se eliminan valores nulos, se fusionan datos y se seleccionan variables para el modelado.

- **Selección de datos:** elegir los conjuntos de datos a utilizar y documentar las razones de inclusión/exclusión.
- **Limpieza de datos:** corregir, imputar o eliminar valores erróneos para evitar errores.

- **Construcción de datos:** derivar nuevos atributos útiles, como calcular el índice de masa corporal a partir de altura y peso.
 - **Integración de datos:** combinar datos de múltiples fuentes para crear nuevos conjuntos de datos.
 - **Formato de datos:** Ajuste la estructura de los datos según se requiera. Por ejemplo, transforme valores de texto que contienen información numérica en formatos numéricos reales, permitiendo así la realización de cálculos matemáticos. Este proceso asegura que los datos estén en el formato adecuado para su análisis posterior.
- 4) **Modelado:** Se elige el modelo adecuado para el negocio y se definen las variables que se utilizarán en el proceso de construcción del modelo.
- **Seleccionar técnicas de modelado:** Es necesario determinar qué algoritmos utilizar para el modelado de los datos, por ejemplo, regresión o redes neuronales.
 - **Generar un diseño de prueba:** Según la estrategia de modelado elegida, puede ser necesario separar los datos en tres subconjuntos: uno para entrenar el modelo, otro para probarlo y un tercero para validar su rendimiento. Esta división permite una evaluación más robusta del modelo.
 - **Construir el modelo:** Aunque suena glamoroso, construir el modelo puede ser tan simple como ejecutar algunas líneas de código como "reg = LinearRegression (). Fit (X, y)".
 - **Evaluar el modelo:** Es importante evaluar adecuadamente el modelo para determinar si cumple con los objetivos empresariales y los criterios de éxito.

5) **Evaluación:** El modelo se prueba y se evalúa si el resultado corresponde a la expectativa del proyecto. Se hacen cambios necesarios como corrección de datos o tratamiento de atributos.

- **Evaluar resultados:** Se revisa si los modelos cumplen los criterios de éxito del negocio y se decide cuál(es) se aprueban para su uso.
- **Revisar el proceso:** Se revisa el trabajo realizado para asegurarse de que se hayan realizado todas las etapas correctamente y no se haya pasado por alto nada. Si es necesario, se corrigen errores.
- **Determinar los siguientes pasos:** En función de los resultados obtenidos, se decide si se procede a la implementación, se iteran más los modelos o se inician nuevos proyectos.

6) **Implementación (deployment):** El modelo desarrollado se implementa en un entorno operativo, generalmente alojándolo en plataformas de nube o en los servidores internos de la organización. Esto permite que el modelo esté disponible para su uso y acceso en situaciones reales del negocio.

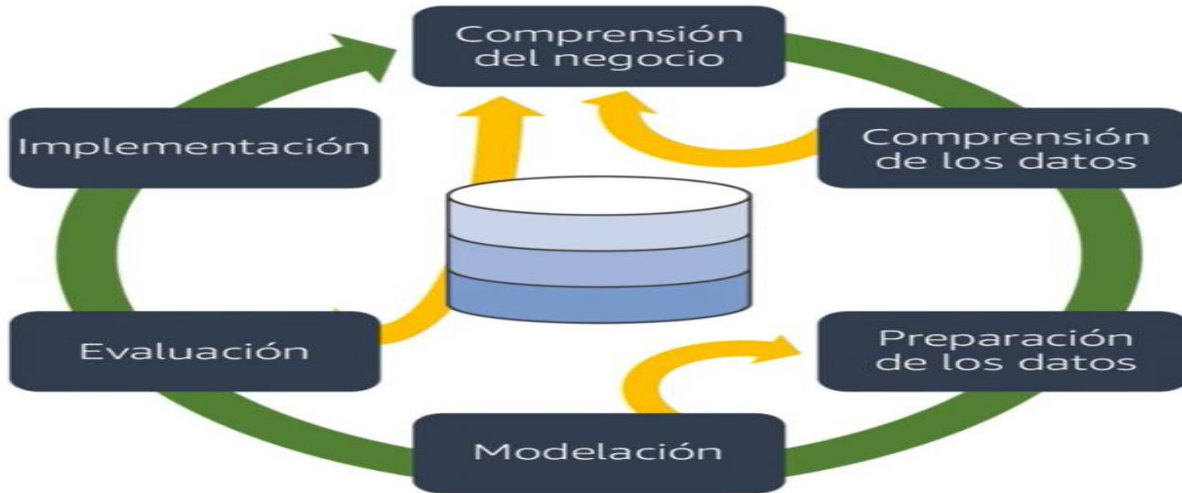
- La fase de implementación puede variar en complejidad dependiendo de los requisitos del proyecto, y puede ser desde la generación de un simple reporte hasta la implementación de un proceso de minería de datos repetible en toda la empresa

(Amazon Web Services, 2023; Caio, 2022; HOTZ, 2023; Jiménez & Merino, 2023)

A manera de resumen se puede ver los pasos de CRISP-DM en la **Error! Reference source not found..**

Figura 4

Fases de CRISP-DM.



1.3.3 Metodología de desarrollo Kanban

Kanban como mencionada (Atlassian, 2023) puede ser una metodología eficaz de gestión del trabajo para un proyecto individual o pequeño. Kanban hace hincapié en el desarrollo continuo e intenta equilibrar el trabajo necesario con la disponibilidad de cada miembro del equipo. Este sistema se basa en tableros Kanban para la gestión visual de proyectos. Cada columna del tablero simboliza una fase del trabajo, como "Trabajo pendiente", "En curso" y "Finalizado".

Las tareas se desplazan de una columna a la siguiente hasta que están terminadas. Kanban funciona bien en equipos de todos los tamaños, sobre todo en equipos remotos. Los tableros Kanban permiten a los miembros del equipo visualizar el trabajo y mantenerse al día independientemente de dónde trabajen. Kanban, en general, se centra en las tareas más importantes del proyecto manteniendo un marco claro (ASANA, 2022).

Existen 4 principios que se debe seguir para implementar Kanban, según (Simon Buehring, 2021):

- a) Empieza con lo que haces ahora;
- b) Comprométete a buscar e implementar cambios progresivos y evolutivos;
- c) Respeta los procesos, los roles y las responsabilidades actuales;
- d) Impulsa el liderazgo en todos los niveles.

Por otro lado, Kanban sigue 6 buenas prácticas que según el equipo de (Teamhood, 2022) son:

a) Visualizar el trabajo; b Limitar el trabajo en curso; c) Gestionar el flujo de trabajo; d) Implementar políticas de procesos explícitas; e) Implementar ciclos de comentarios; f) Mejorar colaborando y evolucionar experimentando.

1.3.4 Aplicación Web

El enfoque propuesto consiste en la detección temprana de malezas mediante el uso de imágenes capturadas por drones y un sistema de predicción basado en un modelo de Deep Learning. Este sistema se implementará en una aplicación web que alojará un modelo pre-entrenado de la arquitectura SSD-VGG, diseñado para analizar las imágenes proporcionadas por el dron. De esta forma, se podrá calcular con eficiencia y precisión la densidad de malezas presentes en la plantación.

Se implementará el proceso mediante la ejecución secuencial de las siguientes etapas.

1.- Se creará un conjunto de imágenes capturadas por drones, las cuales serán utilizadas para el entrenamiento del modelo.

2.- Estas imágenes serán adecuadamente etiquetadas para facilitar el aprendizaje del modelo.

3.- Una vez completado el entrenamiento del modelo SSD-VGG y asegurada su precisión, se desarrollará una aplicación web.

4.- Esta aplicación se desplegará utilizando Gradio, una biblioteca de Python que permite crear interfaces de usuario intuitivas para modelos de aprendizaje automático. El despliegue se llevará a cabo en Google Colab, un entorno de notebooks de código abierto basado en la nube que facilita el desarrollo y la ejecución de modelos de aprendizaje profundo, (Gradio, 2024).

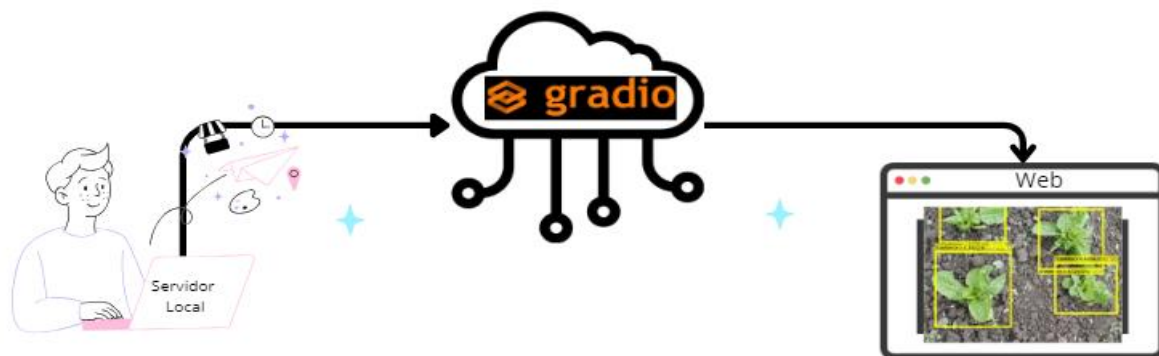
5.- Posteriormente, se ingresarán nuevas imágenes al aplicativo para ser procesadas por el modelo entrenado, con el fin de identificar la cantidad de malezas presentes.

6.- Finalmente, se calculará la densidad de malezas en la plantación utilizando la información proporcionada por el análisis de las imágenes, y se determinará la eficiencia y beneficios de este enfoque en comparación con los métodos tradicionales.

Entonces, se establece un proceso estructurado para el análisis de las imágenes obtenidas por drones y la identificación de malezas en la plantación, utilizando un modelo de Deep Learning basado en la arquitectura VGG.

Figura 5

Despliegue del modelo en Gradio.



1.3.5 Norma ISO 25010

Al finalizar este trabajo se verificará el funcionamiento del modelo de *Deep Learning* en una página web usando la norma ISO/IEC 25010, para medir los criterios de satisfacción del usuario.

Esta norma establece una metodología para evaluar la calidad del software y de los sistemas de información mediante la evaluación de su calidad interna, externa y operativa. En particular, se centra en la calidad de los requisitos del software y de los sistemas de información, y ofrece criterios para evaluarlos. La norma también explica cómo utilizar los criterios de calidad para analizar las necesidades de software y sistemas de información, así como la forma de llevar a cabo la evaluación de la calidad asociada. En conclusión, la norma ISO/IEC 25010 es una herramienta valiosa para garantizar que los requisitos de software y sistemas de información sean de alta calidad y satisfagan las necesidades y expectativas de los usuarios finales (ISO, 2014).

1.4 Artículos Relacionados

- a) El artículo de (Sahoo et al., 2024) menciona que desarrollaron MOD-CVS, un modelo de detección de objetos en movimiento para escenas de video complejas. Utilizaron una arquitectura codificador-decodificador basada en VGG-19 modificado, con un marco de agrupación de características (FPF) y transferencia de aprendizaje. El modelo se probó en conjuntos de datos desafiantes como CDNet 2014, Wallflower y Star dataset, comparándose con 36 técnicas del estado del arte. MOD-CVS logró resultados superiores, con una precisión promedio de 0.8550, recall de 0.8021, F-measure de 0.8269 y un porcentaje de clasificación errónea de 0.8788 en CDNet 2014. El modelo también demostró un rendimiento superior en configuraciones no vistas y una velocidad de procesamiento de 21 cuadros por segundo, superando a otros métodos en precisión y eficiencia computacional.
- b) (Saleh et al., 2024) proponen un enfoque semi-supervisado para la detección de malezas en cultivos. Utilizaron una arquitectura U-Net con un codificador EfficientNetB0 pre-entrenado en ImageNet. El método se basa en un aprendizaje activo iterativo, donde un conjunto inicial de 100 imágenes etiquetadas se amplía gradualmente con 50 imágenes

adicionales en cada iteración. Emplearon técnicas de aumento de datos y un umbral de confianza para seleccionar nuevas imágenes para etiquetar. El conjunto de datos consistió en 5000 imágenes de cultivos de zanahoria, con una división de 60% para entrenamiento, 20% tanto para validación y pruebas. Los resultados mostraron que con solo 300 imágenes etiquetadas (6% del conjunto total), el modelo alcanzó un IoU de 0.8516 y un F1-score de 0.9197, comparable al rendimiento del modelo entrenado con el conjunto completo de datos etiquetados. Este enfoque demostró ser eficiente en términos de tiempo y recursos de etiquetado, manteniendo un alto rendimiento en la detección de malezas.

- c) (Seiche et al., 2024) desarrollaron un sistema de cámara multiespectral de bajo costo (LCS) para la detección de malezas, comparándolo con el sistema de alta gama MicaSense Altum. Utilizaron UAVs para capturar imágenes en campos de maíz, empleando un total de 75 imágenes por sensor, que se procesaron y anotaron para entrenar un modelo de segmentación semántica U-Net. El procesamiento incluyó la normalización de imágenes y la creación de máscaras de vegetación usando el NDVI. Entrenaron el U-Net con una arquitectura ResNet-50 y lo evaluaron mediante índices Jaccard, precisión, recall y F1-score. Los resultados mostraron una precisión general alta: 93.3% para el LCS y 96.8% para el sistema Altum, con una F1-score para detección de malezas de 76% y 82%, respectivamente. Aunque el LCS presentó algunas dificultades con malezas pequeñas y superposiciones, demostró ser una opción viable y económica para la detección de malezas en agricultura de precisión.
- d) (J. Li et al., 2024) desarrollaron el modelo YOLOv7-FWeed para la detección de malezas en campos de soya, mejorando la versión YOLOv7 original. Utilizaron UAVs para capturar imágenes de alta resolución en diferentes etapas de crecimiento de la soya y prepararon un conjunto de datos anotado de 500 imágenes, que aumentaron a 3000 mediante técnicas de mejora de imagen. Entrenaron el modelo en este conjunto de datos

mejorado y evaluaron su rendimiento utilizando métricas como precisión, recall, F1 y mAP. Los resultados mostraron que YOLOv7-FWeed superó a YOLOv7 y YOLOv7-enhanced en todas las métricas, con una precisión de 0.9496, un recall de 0.9125, un F1 de 0.9307 y un mAP de 0.9662. Además, evaluaron la eficacia de la reducción de herbicidas y encontraron que una reducción del 10% combinada con pulverización electrostática controlaba eficazmente el crecimiento de malezas sin afectar el crecimiento de la soya, lo que se reflejó en un aumento constante del área foliar y el peso seco de la planta.

- e) (García-Navarrete et al., 2024) realizaron una revisión sistemática sobre el uso de redes neuronales convolucionales (CNN) para la detección e identificación de malezas en cultivos. Analizaron 61 artículos publicados entre 2019 y 2023, siguiendo las directrices PRISMA. El estudio se enfocó en identificar las arquitecturas CNN y tecnologías utilizadas. Para ello, emplearon las bases de datos Web of Science y Scopus. Los resultados mostraron que las cámaras digitales fueron la tecnología más utilizada (49.2%), seguidas por cámaras en drones (29.5%) y smartphones (11.5%). En cuanto a las arquitecturas CNN, la familia YOLO fue la más adoptada, seguida por VGG, ResNet, Faster R-CNN, AlexNet y MobileNet. Los autores encontraron que en los últimos dos años hubo un crecimiento masivo en el número de artículos publicados sobre detección de malezas, atribuido al desarrollo de arquitecturas CNN más eficientes, mayor capacidad de procesamiento y reducción en el precio de cámaras y unidades de procesamiento gráfico. El estudio concluye que esta revisión proporciona una actualización sobre CNNs que servirá como punto de partida para investigadores que deseen implementar técnicas de detección e identificación de malezas.
- f) En el estudio de (Dang et al., 2023) analiza el desarrollo de sistemas de visión artificial para el control automatizado de malas hierbas, debido a la evolución de malas hierbas resistentes y la preocupación por la salud y el

medio ambiente. Los autores presentan un nuevo conjunto de datos (CottoWeedDet12) de 12 tipos de malas hierbas específicas para la producción de algodón en el sur de Estados Unidos, junto con una lista de 25 detectores de objetos YOLO de última generación para la detección. Todos los modelos YOLO, especialmente YOLOv5n y YOLOv5s, han mostrado un gran potencial para la detección de malas hierbas en tiempo real, y el aumento de datos podría aumentar la precisión de la detección de malas hierbas.

1.5 Preguntas de Investigación

En la Tabla 3 se puede observar que algunas preguntas de investigación.

Tabla 3

Preguntas de investigación

Pregunta	Articulo	Respuesta
QR1. ¿Cómo afinar un modelo de DL para mejorar el rendimiento?	(Espejo-Garcia et al., 2020; Mahmudul Hasan et al., 2022)	Se puede afinar con: Aumento de datos; Cambio de contraste;Desenfocue; Rotación Gaussian.; <i>Noise; Fine-tuning</i>
QR2. ¿Qué arquitectura de CNN se usan para la detección automática de malas hierbas?	(Mahmudul Hasan et al., 2022; Moazzam et al., 2021)	Arquitecturas: Region Proposal Network, Fully Convolutional Network, Graph Convolutional network, hybrid network;

		VGG-Beet; Mask R-CNN, ResNet-101
QR3. ¿Cómo crear y anotar un conjunto de datos propio?	(Mahmudul Hasan et al., 2022)	<p>Crear: Eliminación del fondo, el cambio de tamaño de la imagen, la segmentación de componentes verdes, la eliminación del desenfoque de movimiento, la eliminación de ruido, la mejora de la imagen, la extracción de índices de vegetación de color y el cambio de modelo de color.</p> <p>Anotación: En cuanto al etiquetado de datos, se pueden utilizar diferentes técnicas, como anotaciones de cuadros delimitadores, anotaciones de píxeles y anotaciones de nivel de imagen. También es posible</p>

		<p>utilizar técnicas semi-supervisadas o semiautomáticas para reducir los costos de anotación de datos y generar grandes conjuntos de datos etiquetados.</p>
<p>QR4. ¿Qué hardware específico o servicios en la nube resulta adecuado para entrenar modelos de DL?</p>	<p>(Espejo- et al., 2020; Subeesh et al., 2022)</p>	<p>Hardware:</p> <p>Ubuntu 18.04, CPU de 16 núcleos (Ryzen Threadripper 1950x), 64 GB de ram DDR4 y una GPU GeForce RTX 2080Ti con 11 GB de memoria DDR5 integrada, Cuda</p>

<p>QR5. ¿Qué métricas y gráficas se usan para validar el rendimiento de los modelos de DL?</p>	<p>(Espejo-Garcia et al., 2020; J. Li et al., 2024)</p>	<p>Métricas:</p> <p>a) F1 score, Esta métrica se utiliza como métrica de evaluación para la clasificación;</p> <p>b) recall es la es la proporción de categorías correctas con respecto al conjunto de datos original</p> <p>c) AP(Average precisión) (es la precisión media de las categorías individuales</p> <p>d) mAP(mean AP) es la media del AP de todas las categorías.</p>
--	---	---

<p>QR7. ¿Cuál es la altura y velocidad adecuada del dron para capturar imágenes y regiones de interés (ROIs) de alta resolución (px/cm)?</p>	<p>(Hasan et al., 2021; Moazzam et al., 2021)</p>	<p>Altura: (2, 2.5 10, 15 y 20) m</p>
<p>QR8. ¿En qué bandas del espectro electromagnético (visible, infrarrojo, ultravioleta) están trabajando?</p>	<p>(Mahmudul Hasan et al., 2022; Moazzam et al., 2021)</p>	<p>Espectro de Banda: Analizaron las bandas espectrales visibles y cercanas al infrarrojo para la clasificación de malezas, mientras que Farooq et al. (2018b) utilizaron imágenes hiperespectrales en el rango visible e infrarrojo cercano para la clasificación de malezas.</p>

		B + CIR + G + NDVI + NIR + R + RE
QR9. ¿Qué espacio de color y formato de imágenes resulta adecuado?	(Hasan et al., 2021)	RGB
QR10. ¿Qué cultivos y tipos de malezas se están detectando y dónde?	(Espejo-Garcia et al., 2020; Moazzam et al., 2021; Subeesh et al., 2022)	Cultivos: maíz, remolacha azucarera; tomate y el algodón; Maleza: belladona negra y la hoja de terciopelo, Grecia Cultivo: pimiento, India

Capítulo 2

Desarrollo del proyecto

El objetivo principal del proyecto es desarrollar la detección automática de malezas en cultivos de papa a través de imágenes tomadas por un dron. Para crear un modelo de detección preciso, se utilizará la arquitectura de la red neuronal convolucional VGG. La metodología CRISDM se utilizará como guía para el proyecto, que incluirá las etapas de comprensión, recolección, preparación, modelado, implementación y evaluación de datos. La metodología Kanban se utilizará para supervisar el progreso del proyecto y garantizar una gestión eficiente y transparente del flujo de trabajo. El etiquetado de imágenes se realizará utilizando Roboflow, una plataforma que se especializa en la anotación y preparación de datos para el entrenamiento del modelo. Finalmente, el modelo que se ha creado se desplegará en Gradio, una herramienta que facilita la creación de interfaces de usuario para modelos de aprendizaje automático, lo que facilita su acceso y uso.

2.1 Metodologías y herramientas

2.1.1 Metodología CRISP-DM

Según (Krishnaswamy et al., 2023) Dentro del campo de la minería de datos, el enfoque CRISP-DM (Cross Industry Standard Process for Data Mining) ha ganado reconocimiento como un estándar influyente en el mercado.

(Amazon Web Services, 2023) describe que la metodología CRISP-DM establece un proyecto de minería de datos como una secuencia de fases:

1. Comprensión del negocio.

Es la etapa inicial y fundamental que implica entender el ámbito agrícola, los retos relacionados con la detección de malezas, las demandas de los agricultores y el potencial de la tecnología para abordar esta problemática.

2. Compresión de los datos.

En esta fase se define con que datos se va a trabajar, en este caso imágenes de cultivo de papas con presencia de maleza que serían los objetos para detectar con el modelo cnn, y también la altura desde donde el dron va a capturar las imágenes.

3. Preparación de los datos.

En esta parte es donde ya se cuenta con el conjunto de datos completo para el respectivo preprocesamiento, donde se realizará recortes, eliminación de ruidos, cambio de color RGB a escala de grises entre otros pasos importantes para mejorar el conjunto de datos.

4. Modelado.

Implica el desarrollo y la implementación de algoritmos del modelo de detección de objetos usando la arquitectura base VGG, también se refiere a las técnicas de aprendizaje profundo para entrenar modelos capaces de identificar y clasificar eficientemente las malezas en las imágenes.

5. Evaluación.

En esa fase se realizó algunas evaluaciones como evaluación de métricas cuantitativas específicamente para la detección de objetos como mAP(mean average precision), IOU(Intersection over unión), también lo que fue los análisis de errores y comparación con otros modelos

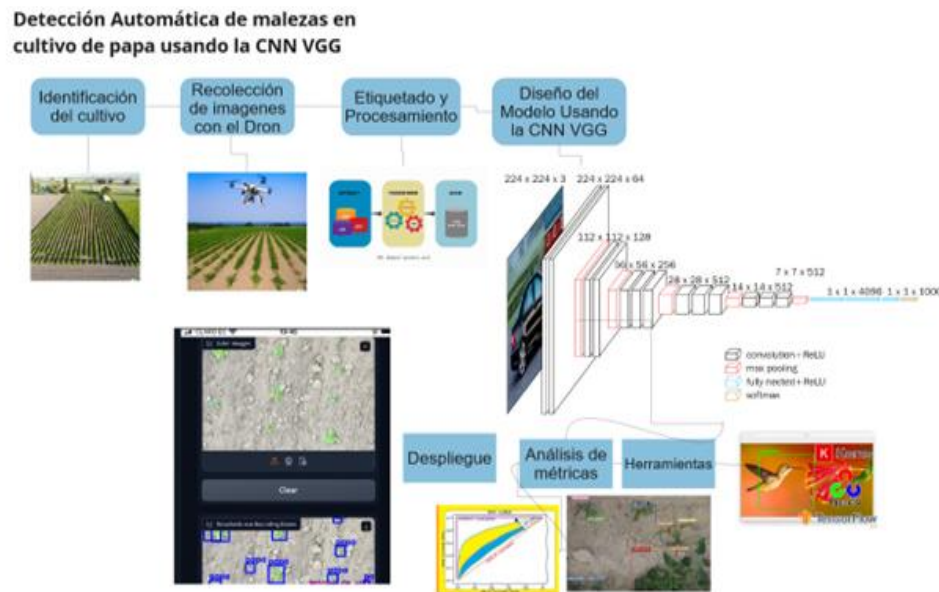
6. Despliegue.

Para el despliegue del modelo en un entorno de producción, se buscó algunas alternativas de alojamiento como: AWS, Azure, Google Cloud y Gradio. En donde se puede probar el funcionamiento del modelo.

Ver la Figura 6, para observar de manera grafica el proceso que se empleó en esta investigación.

Figura 6

Proceso CRISDM del proyecto.

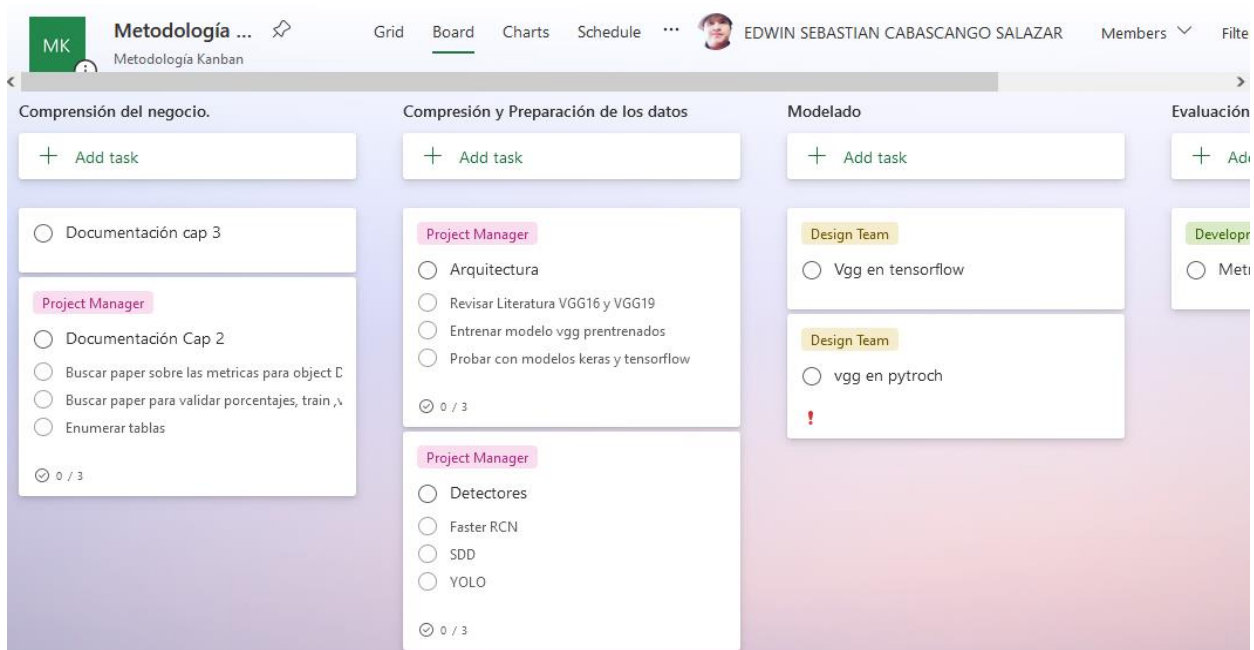


2.1.2 Metodología Kanban

En este proyecto, también se empleó la metodología Kanban, la cual fue utilizada para gestionar y supervisar el proceso de desarrollo de manera efectiva. Kanban es una metodología que permite visualizar y controlar el flujo de trabajo, lo que facilita la identificación de tareas pendientes, la asignación de recursos y la mejora continua del proceso (Atlassian, 2023). Para implementar Kanban, se utilizó la herramienta Planner de Microsoft (Microsoft, 2023), que permitió monitorear el progreso del proyecto, identificar áreas de mejora y asegurar que las tareas se completaran de manera oportuna, al mismo tiempo que se refinaban y pulían los detalles del proyecto, ver Figura 7.

Figura 7

Tablero Kanban para gestión de proyectos.



2.1.3 Características del Dron del Dron

Para el paso 2 de CRIS-DM es decir para la recopilación de imágenes se utilizó un **dron DJI Mavic 2 Pro**. A continuación, se presenta las especificaciones en la Tabla 4 más relevantes del dron que se puede apreciar en la Figura 8, usado en esta investigación.

Tabla 4

Especificaciones del Dron.

Componente	Información
Sensor	1" CMOS Píxeles efectivos: 20 millones
Lente	FOV: alrededor de 77 °

Tamaño de imagen fija	5472x3648
Resolución de video	4K: 3840x2160 24/25/30p 2.7K: 2688x1512 24/25/30/48/50/60p FHD: 1920x1080 24/25/30/48/50/60/120p
Tasa de bits de video máxima	100Mbps
Sistema de archivos compatible	FAT32 (≤ 32 GB) exFAT (> 32 GB)
Formato de foto	JPEG / DNG (RAW)
Formato de video	MP4 / MOV (MPEG-4 AVC/H.264, HEVC/H.265)

Nota. (DJI, 2021)

Figura 8

Dron DJI Mavic 2 Pro



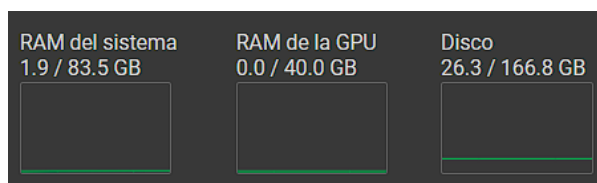
Nota. Tomado de *DJI: MAVIC 2 Specs* por (DJI, 2021).

2.1.4 Herramientas de desarrollo

Para el desarrollo del modelo, se utilizó la plataforma *Kaggle* que tiene una interfaz Jupyter Notebooks personalizable y sin configuración que permite acceder de manera gratuita a GPUs (Data Science, 2023). Paralelo a esto cuando se requirió mayor poder de cómputo se utilizó Google Colab, ver Figura 9.

Figura 9

Recursos Colab Pro



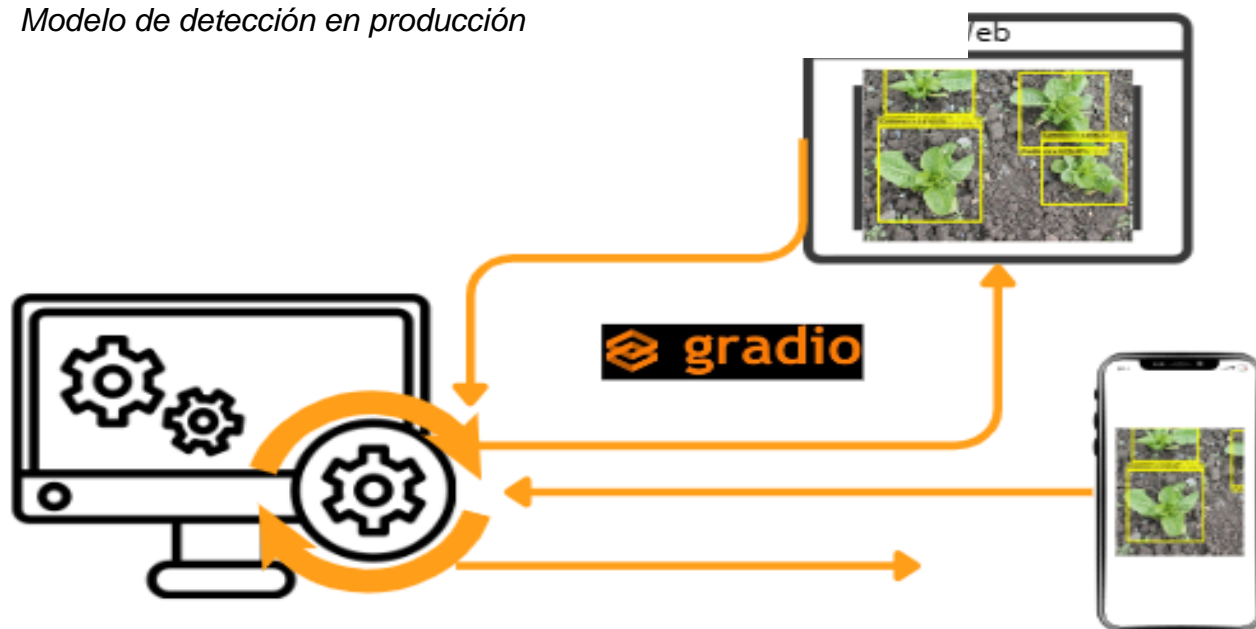
Debido a que las imágenes originales tenían una resolución grande de 5742 x 3648 pixeles, se utilizó *Visual Studio Code* para crear scripts que permitieran cortar las imágenes y sus etiquetas en tamaños específicos.

Una vez creado el modelo se utilizó Gradio que es una plataforma de código abierto que facilita el desarrollo y la difusión de aplicaciones de aprendizaje automático. Su interfaz fácil de usar y la posibilidad de crear aplicaciones sin

tener habilidades de programación avanzadas son sus principales ventajas. Para mejorar el rendimiento de los modelos de aprendizaje automático, permite compartir aplicaciones mediante enlaces, personalizarlas fácilmente y realizar pruebas en tiempo real (Gradio, 2024) ver Figura 10.

Figura 10

Modelo de detección en producción



Nota. Elaboración propia.

Finalmente, se empleó la Norma ISO 25010 para garantizar la calidad del software.

En la Tabla 5 se puede observar a modo resumen la función de cada herramienta

Tabla 5

Detalles de las herramientas de desarrollo

Herramienta	Función
Dron	Recolección del dataset

Visual Studio Code y Python	Herramienta para la creación de scripts para transformaciones de imágenes.
Planner	Herramienta para la gestión de la metodología Kanban
Kaggle	Plataforma online con GPUS gratuitas
Google Colab pro	Entorno con mayor capacidad computacional para entrenar modelos.
Norma ISO 25010	Proporcionar reglas de la calidad del software

Nota. Elaboración propia.

2.2 Construcción del conjunto de imágenes

2.2.1 Recopilación de imágenes

Los datos para este estudio fueron obtenidos de manera meticulosa y sistemática, asegurando su representatividad y calidad óptima. Se seleccionaron diversas plantaciones de papas en un rango de edad de 2 a 5 semanas, ver la Figura 11, ya que durante este lapso se observa claramente tanto el crecimiento de las plantas de papa como la presencia de malezas. Esta selección se realizó con el objetivo de garantizar la visibilidad adecuada de ambos elementos en las imágenes recolectadas.

Figura 11

Planta de papa a 4 semanas de crecimiento.



Nota. Elaboración propia.

Además, se prestó mucha atención a la selección de los terrenos para asegurarse de que fueran ligeramente planos ver Figura 12. Esta medida es fundamental para garantizar que los píxeles en las imágenes obtenidas mantengan un tamaño uniforme en todas las tomas y áreas de la plantación. Este método de precaución garantiza que los datos de entrenamiento sean cohesivos y reflejen las condiciones del terreno (Robayo & García, 2022).

Figura 12

Plantación de papa.



Nota. Elaboración propia.

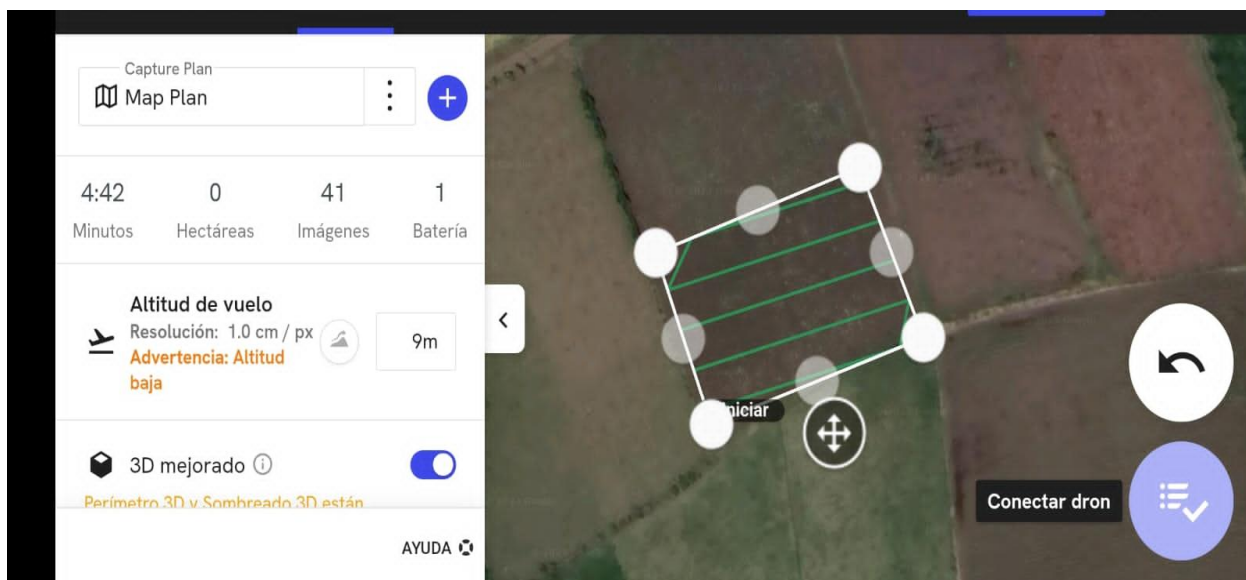
Las plantas seleccionadas se encuentran en el Cantón Espejo y Montufar. Las imágenes se tomaron entre las 10 y las 2 de la tarde para mostrar varios tipos de iluminación a 9 m de altitud y a una velocidad de 1 m/s, que son los valores mínimos que se pueden programar en el software de control de vuelo de Dron Deploy. Esta altura resulta en una resolución de 1.0 cm/px.

Para utilizar DroneDeploy, primero debe crear un nuevo proyecto y definir el área de interés donde se tomarán los pasos. La aplicación muestra una ventana de Google Maps que le permite localizar el punto de referencia del vuelo. Después de configurar esta sección, puede ajustar el área de captura para adaptarse al terreno deseado. La interfaz muestra un polígono que define el área de vuelo, junto con la información y las opciones de configuración en el lado derecho, y un botón en la esquina inferior derecha para iniciar el vuelo y conectar el dron, (Robayo & García, 2022).

La representación gráfica mostrada en la Figura 13 ilustra tanto la ruta seguida por el dron como la configuración del proyecto.

Figura 13

Trayectoria y configuración del proyecto.



El área seleccionada para la recolección de muestras de cultivo se encuentra en algunas propiedades, en la Provincia del Carchi. La siembra de los cultivos tuvo lugar el 7 de mayo de 2023, y el proceso de monitoreo y recolección de muestras se inició alrededor del 20 de mayo. Se decidió esperar aproximadamente dos semanas después de la siembra para que las plantas alcanzaran un tamaño considerable y pudieran ser visualizadas adecuadamente desde el aire, para mayor detalle ver la Tabla 6.

Tabla 6

Fechas en las que se capturaron las imágenes en los terrenos elegidos

Terreno	Fecha
Terreno 1 - San Gabriel, comunidad de Chutan Bajo, 77°51'03"W (Propiedad del padre del Ing. Marco Pusdá).	20/05/2023.
	28/05/2023.
	04/06/2023.
Terreno 2 - San Gabriel, 77°48'40"W (Propiedad anónima).	04/06/2023.
Terreno 3 – Cuesaca, Avenida de la Prehistoria, 77°52'40"W (Propiedad del tío del estudiante Kevin Vinueza).	17/06/2023.
Terreno 4 - El Ángel, barrio San Vicente, 0°38'21"N (Propiedad anónima).	16/12/2023.

Terreno 5 - El Ángel, barrio
San Vicente, 77o55'59"W
(Propiedad anónima). 16/12/2023.

Terreno 6 - El Ángel, barrio
San Vicente, 77o55'59"W
(Propiedad anónima). 16-12-2023.


Terreno 7 - El Ángel, barrio
San Vicente, 0o38'13"N
(Propiedad anónima). 16-12-2023.






Nota. Elaboración propia.

Además de la recopilación de imágenes, se capturaron fotografías de las malezas encontradas en el área para su posterior clasificación: kikuyo (*Pennisetum clandestinum*), lengua de vaca (*Rumex crispus*), diente de león (*Taraxacum officinale*) y otros (plantas con muy pocas muestras) ver Tabla 7. Asimismo, se fotografiaron las hojas de las plantas de papa. Estas fotografías fueron tomadas en tres momentos distintos, con una separación de una semana entre cada sesión, lo que proporciona datos para examinar la evolución y el desarrollo de las plantas y las malezas a lo largo del tiempo (EDIFARM & CÍA, 2018).

Tabla 7

Planta y malezas de la presente investigación

Nombre de la planta	Imagen
Papa	

Lengua de vaca	
Kikuyo	
Diente de León	
Otras	 

En la recopilación de las imágenes se realizó por etapas con la finalidad de tener variabilidad en los datos. Se recopiló un total de 2447 imágenes. En la Tabla 8 se puede observar el número de imágenes capturados por fecha y hora en las distintas propiedades.

Tabla 8

Número de imágenes capturadas por fechas en los terrenos

Terreno	Fecha y hora	Número de imágenes(4K)
Terreno 1 - San Gabriel, comunidad de Chutan Bajo, 77o51'03"W (Propiedad del padre del Ing. Marco PUSDÁ).	20/05/2023, 9:10 - 9:55.	539.
	28/05/2023, 10:07 - 10:42.	409.
	04/06/2023, 11:25 - 12:38.	926.
Terreno 2 - San Gabriel, 77o48'403"W (Propiedad anónima).	04/06/2023, 14:12 - 14:20.	197.
Terreno 3 – Cuesaca, Avenida de la Prehistoria, 77o52'40"W (Propiedad del tío del estudiante Kevin Vinueza).	17/06/2023, 8:58 - 9:20.	209.
Terreno 4 - El Ángel, barrio San Vicente, 0o38'21"N (Propiedad anónima).	16/12/2023, 10:23 - 10:28.	61.
Terreno 5 - El Ángel, barrio San Vicente,	16/12/2023, 10:52 - 10:55.	28.

77o55'59"W (Propiedad anónima).		
Terreno 6 - El Ángel, barrio San Vicente, 77o55'59"W (Propiedad anónima).	16-12-2023, 11:18 - 11:25.	53.
Terreno 7 - El Ángel, barrio San Vicente, 0o38'13"N (Propiedad anónima).	16-12-2023, 11:51 - 11:55.	25.
Total.		2447.

Nota. Elaboración propia.

En la captura de las imágenes se utilizó la cámara digital del dron *DJI Mavic 2 Pro* a una altura de 9 metros sobre el cultivo, con una velocidad de vuelo de 1 metro por segundo, empleando una lente con un campo de visión (FOV) aproximado de 77 grados. Las imágenes se capturaron en una resolución de video 4k, con dimensiones de 5472x3648 píxeles, ver la Tabla 9 donde se detalla los parámetros elegidos para la captura de imágenes.

Tabla 9

Parámetros elegidos para la captura de imágenes

Parámetro	Valor
Altura de vuelo.	9 metros.
Velocidad de vuelo.	1 metro/segundo.
Resolución de imagen.	5472 x 3648 píxeles.

Tiempo entre captura de cada imagen.

4 segundos.

Nota. Elaboración propia.

2.2.2 Preprocesamiento del dataset

Las imágenes capturadas por el dron tienen una resolución superior a 4K (5742x3648) ver Figura 14, pero esta alta resolución no es ideal para nuestro propósito, ya que la arquitectura VGG requiere una entrada sugerida de 224 x 224 (Simonyan & Zisserman, 2014) , aunque puede adaptarse hasta 500 x 500 con ciertas modificaciones. Por lo tanto, es necesario redimensionar las imágenes utilizando un script de Python para este fin. Es importante tener en cuenta que, al reducir el tamaño de la imagen, se puede perder información, aunque el tiempo de entrenamiento disminuya. Para este proyecto, las imágenes se recortaron a tamaños de 400x400, 500x500 y 600x600.

Figura 14

información de imagen original.

Imagen	
Id. de imagen	
Dimensiones	5472 x 3648
Ancho	5472 píxeles
Alto	3648 píxeles
Resolución horizontal	72 ppp
Resolución vertical	72 ppp
Profundidad en bits	24
Compresión	
Unidad de resolución	2
Representación del color	sRGB
Bits comprimidos/píxel	

Durante el proceso de preprocesamiento, se lleva a cabo una depuración de las imágenes para eliminar aquellas que no aportan de manera efectiva al entrenamiento del modelo. Esto incluye imágenes capturadas al inicio y al final del vuelo, cuando el dron está despegando o aterrizando, así como imágenes

con obstrucciones visuales como árboles, pasillos de plantaciones o líneas de cultivo que no siguen una trayectoria paralela al vuelo ver Figura 15 y 16. Asimismo, se descartaron imágenes que presentaban desenfoque o estaban capturadas a alturas muy altas de lo establecido, y así asegurar la calidad del conjunto de datos utilizado en el proceso de entrenamiento.

Figura 15

Obstrucciones visibles en la fotografía.



Figura 16

Imagen con altura que excede el límite establecido



Después de depurar el conjunto de imágenes, se trabajó con un total de 326 imágenes ver Tabla 10.

Tabla 10

Banco de imágenes depuradas.

Criterio de eliminación	Número de imágenes utilizables resultantes(4K)
Eliminación de todas las imágenes adquiridas en el terreno 1 y 3.	364.
Eliminación de aquellas que presentaban desenfoque.	38.
Total.	326.

2.2.3 Etiquetado de datos

El etiquetado de datos es esencial en el desarrollo de modelos de inteligencia artificial al asignar etiquetas o categorías a los datos, permitiendo a los algoritmos de aprendizaje automático identificar patrones y tomar decisiones precisas. La calidad y precisión de los modelos de IA dependen en gran medida de la calidad de los datos de entrenamiento, por lo que este proceso es crucial. Puede realizarse manualmente por humanos o de forma automatizada mediante algoritmos y técnicas de procesamiento de datos. Además, se deben considerar cuidadosamente las implicaciones éticas y de privacidad para garantizar el manejo adecuado de la información sensible y el cumplimiento de las regulaciones aplicables (IMB, 2023).

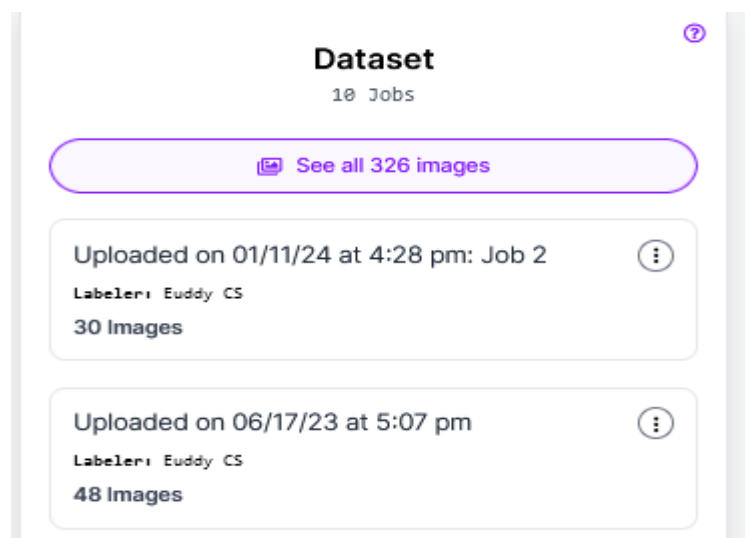
Roboflow es una herramienta de software potente que utiliza inteligencia artificial y aprendizaje automático para permitir a las aplicaciones visualizar e

identificar objetos en imágenes y videos. Su conjunto de funciones incluye un asistente de etiquetado basado en el modelo Segment Anything Model (SAM), que facilita el etiquetado rápido y preciso de imágenes. Además, ofrece herramientas para el entrenamiento de modelos personalizados y permite el despliegue de estos modelos en aplicaciones o sistemas para aprovechar sus capacidades de detección y categorización de objetos. Los pasos para utilizar Roboflow incluyen cargar las imágenes, etiquetar los objetos, entrenar el modelo de visión por computadora y desplegar el modelo en la aplicación o sistema deseado (Roboflow, 2020).

Para empezar con el etiquetado usando la herramienta Roboflow primero se subió todo el conjunto de datos, un total de 326 imágenes, ver Figura 17.

Figura 17

Total de imágenes a etiquetar.



Una vez el conjunto de imágenes en Roboflow se establecieron 5 etiquetas para los objetos presentes en las imágenes Figura 18.

Figura 18

Etiquetas de los elementos a detectar.

Classes

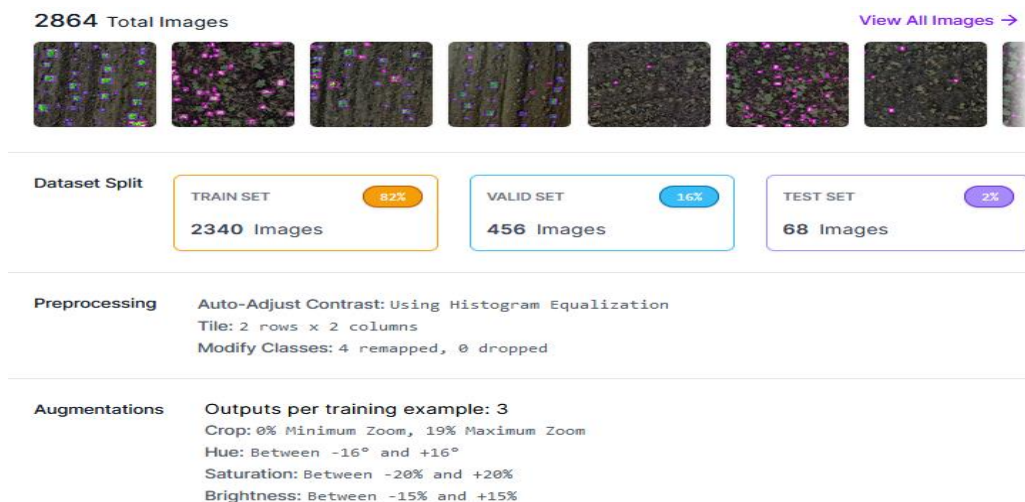
Annotation

Color	Class Name
	Diente-de-leon
	Kikuyo
	lengua de vaca
	otros
	papa

En los trabajos de (Chicaiza, 2022; Sharma et al., 2022), mencionan que se utilizó la herramienta *Roboflow* para realizar las transformaciones necesarias, incluyendo el cambio de tamaño, histogramas de las imágenes. Además, se exploraron las opciones de aumentación, incluyendo la saturación y el balance de luz, con el fin de generar imágenes que aparentaran haber sido capturadas tanto en la mañana como en la tarde del día. El cambio de tamaño se utilizó para reducir el peso y utilizar el tamaño óptimo del modelo seleccionado para el entrenamiento, ver Figura 19.

Figura 19

Detalle final luego del preprocesamiento y aumento de datos en Roboflow.



Una vez finalizado el procedimiento de etiquetado de las fotografías, se observó que hay maneras de descargarla el conjunto de datos en varios formatos. En este caso, se descargó el conjunto de datos en formato *Pascal VOC* Figura 2020. Este formato incluye la imagen, así como un archivo XML, con información específica, como el nombre y el tamaño de la imagen, además de la categorización y las coordenadas *bouding box*. El formato se eligió en función a los requerimientos del modelo (SSD-VGG) utilizado.

Figura 20

Formato Pascal VOC para entrenamiento del modelo.

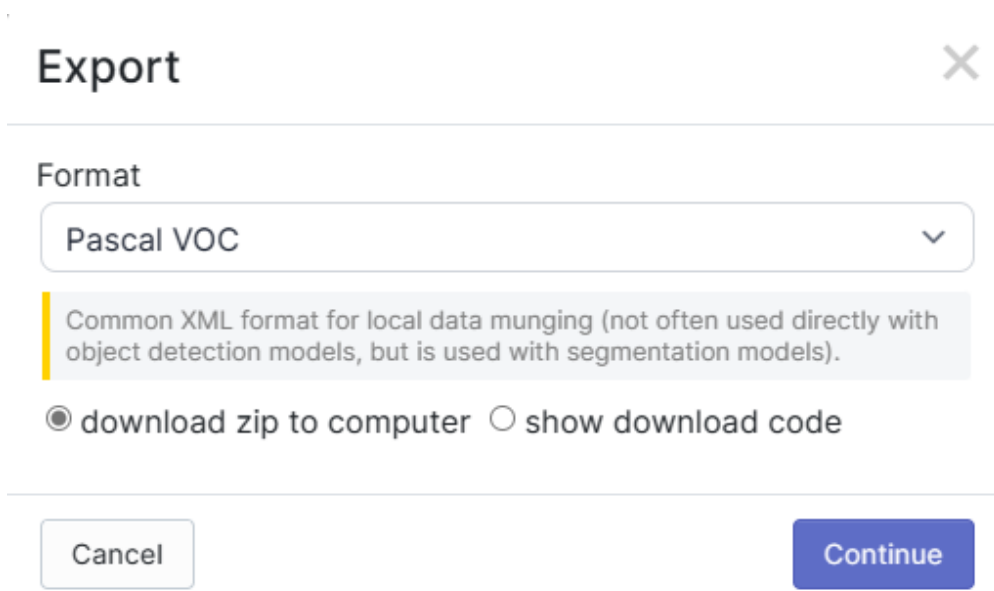
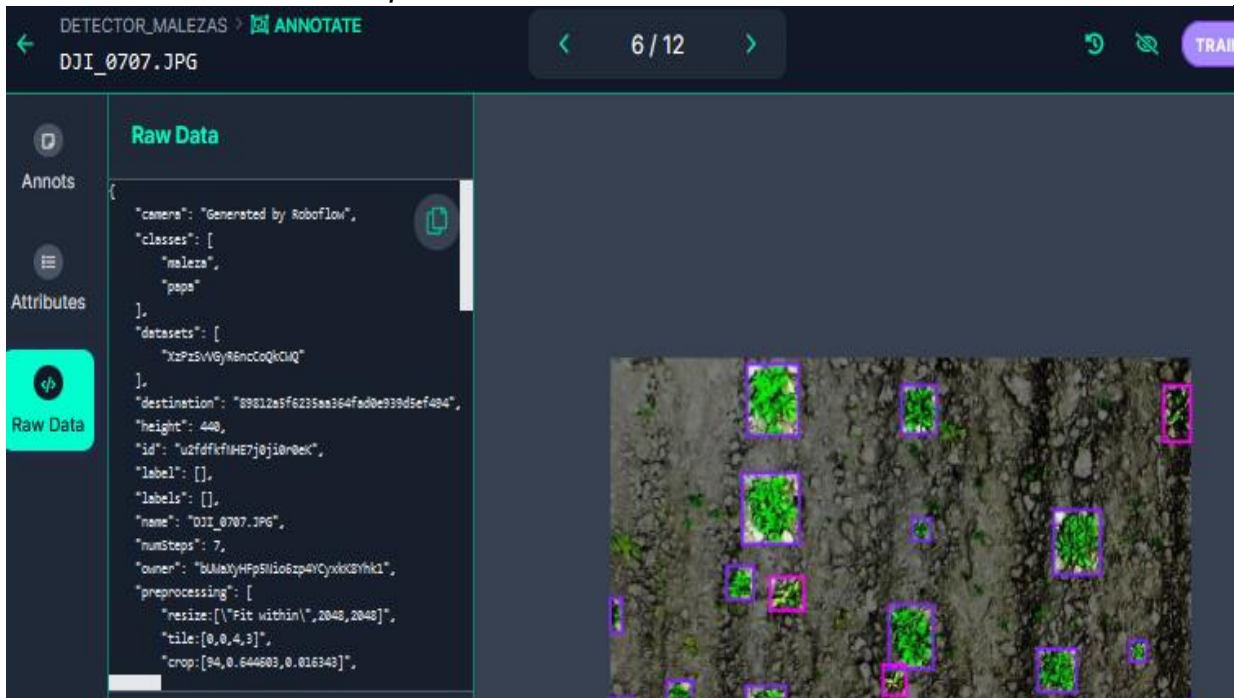


Figura 21

Detalles del archivo de etiquetas XML.

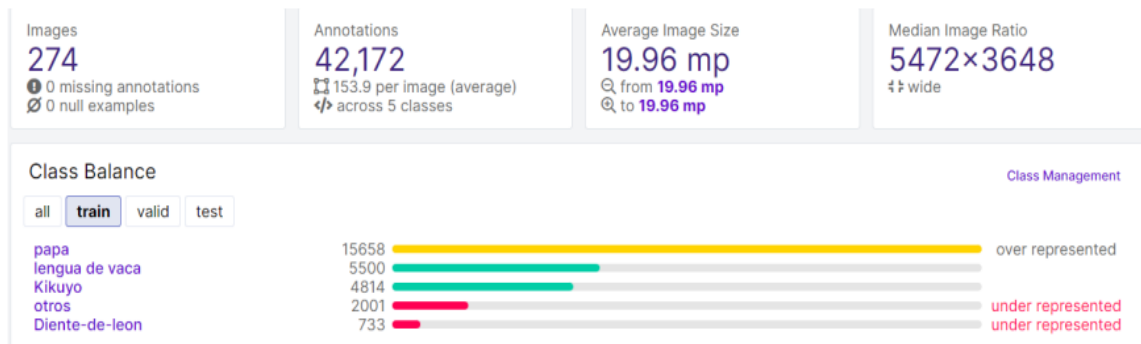


2.3.4 Métricas del conjunto de datos final

El conjunto de datos para las pruebas iniciales fue de 274 imágenes, ver en la Figura 22 **Figura 22**, se puede observar con mayor detalle la información más relevante del conjunto de datos utilizado en este trabajo.

Figura 22

Información descriptiva del conjunto de datos

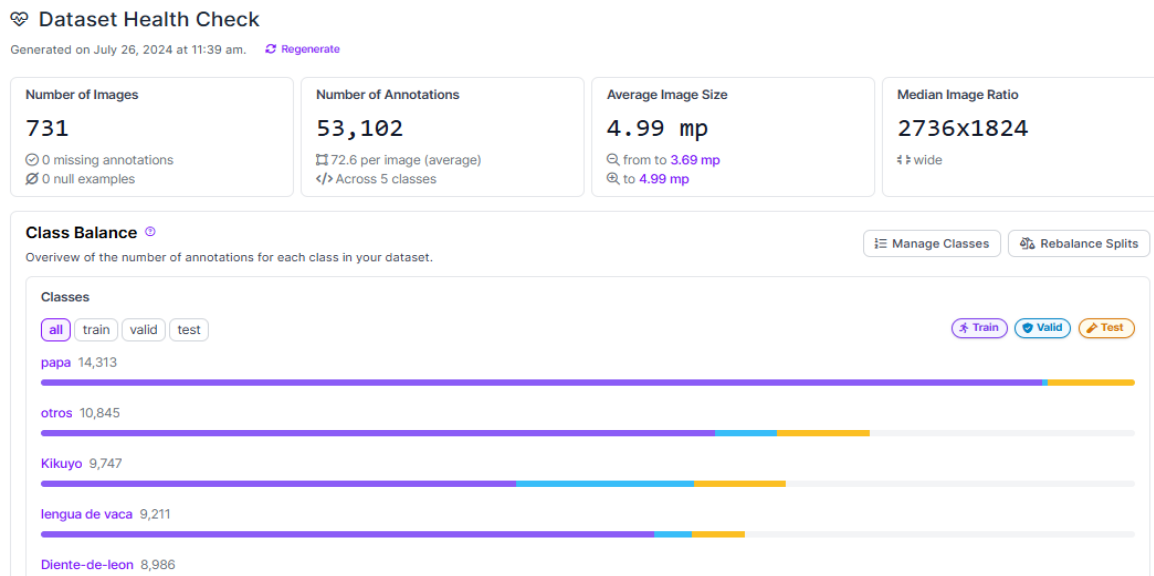


Nota: Resultado obtenido del proyecto UTN (Roboflow, 2024).

El conjunto de imágenes finales fue de 731, ver en la Figura 23.

Figura 23

Estado del conjunto de datos pertenecientes al Dataset #2 “Balanceado”.



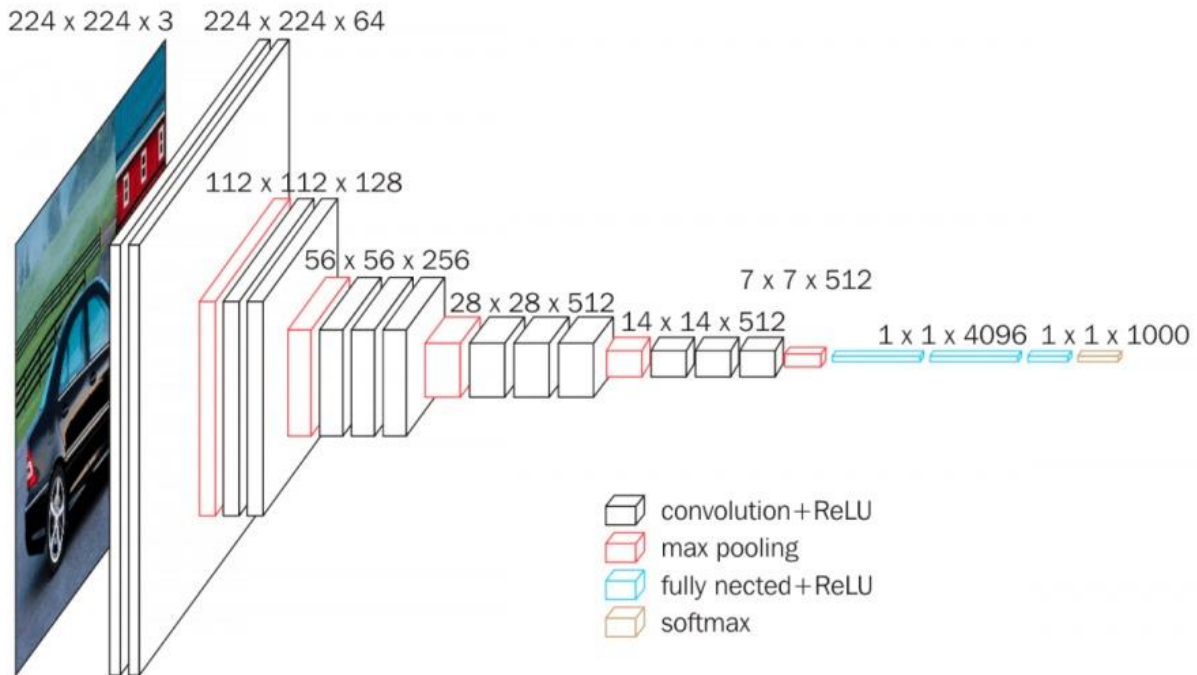
Nota: Resultado obtenido del proyecto UTN (Roboflow, 2024).

2.3 Modelo de detección de objetos basado en VGG

El modelo utilizado en este estudio consta de dos partes, la primera se basa en la arquitectura VGG que se muestra en la Figura 24, propuesta por (Simonyan & Zisserman, 2014), la cual se seleccionó debido a que representa una propuesta de investigación de la Carrera de Software de la Universidad Técnica del Norte, además de demostrar una precisión satisfactoria en la clasificación de objetos al utilizar como base esta red neuronal convolucional

Figura 24

Arquitectura VGG.

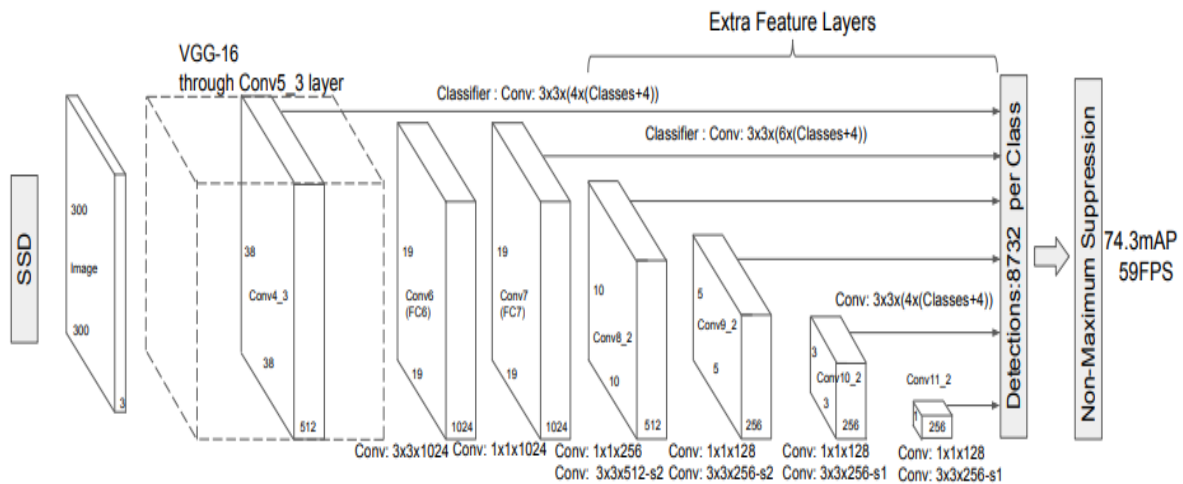


Nota: Adaptado de *UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation*, de Pravitasari Anindya, 2020, Telkomnika (<http://doi.org/10.12928/telkomnika.v18i3.14753>).

La segunda parte del modelo consta de un detector de objetos se optó denominado SSD (Single Shot MultiBox Detector) propuesta por (W. Liu et al., 2016), con una red base VGG, debido a su eficacia probada en la detección de objetos en imágenes. La combinación de SSD con VGG como se puede ver en la Figura 25, ofrece una arquitectura de red profunda que permite la detección precisa y rápida de objetos en tiempo real. Además, la red base VGG (Simonyan & Zisserman, 2014) es conocida por su capacidad para extraer características de imágenes con una alta resolución espacial, lo que resulta beneficioso para la detección de objetos en escenarios complejos como la detección de malezas en imágenes capturadas por drones.

Figura 25

Arquitectura SSD con backbone VGG



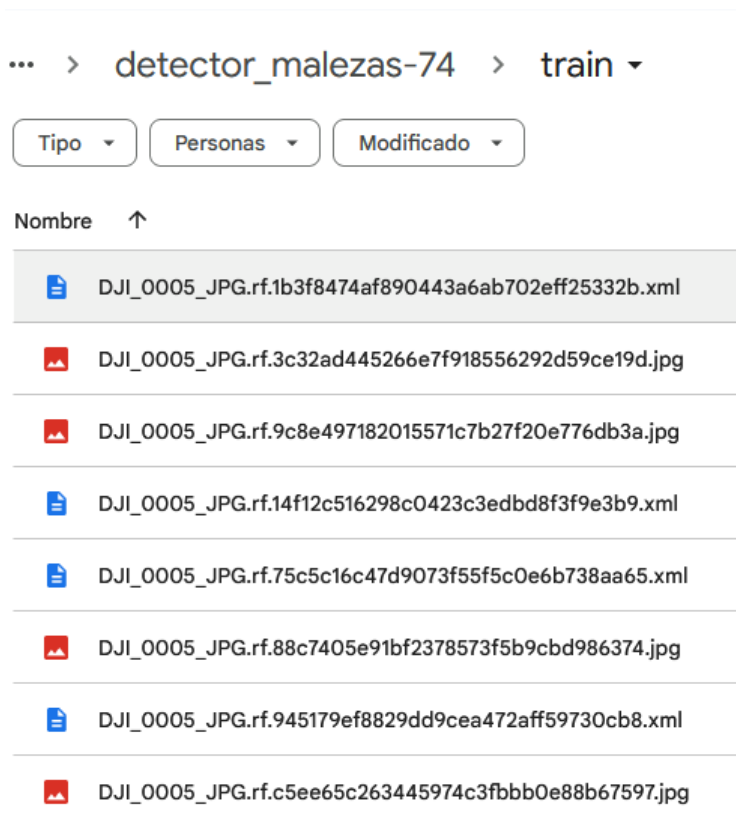
Nota. Adaptado de *Single shot multibox detector*, por (W. Liu et al., 2016), arXiv ([1512.02325 \(arxiv.org\)](https://arxiv.org/abs/1512.02325)).

2.4 Implementación y Evaluación del Modelo

En primer lugar, se descargó el dataset aumentado y dividido desde *Roboflow* en el formato compatible que en este caso *PASCAL VOC* que por cada imagen vienen con sus respectivas etiquetas y cajas delimitadoras (bounding box) en archivo .xml como se ve en la Figura 26.

Figura 26

Estructura de la carpeta "train" que contiene las imágenes junto con sus etiquetas.



Para implementar el modelo, se eligió inicialmente Kaggle como plataforma de entrenamiento debido a sus recursos computacionales robustos y gratuitos, como las GPUs, Los recursos computacionales de Kaggle son 29 GB de RAM y 15 de GPU P100, 73GB de Disco, que se puede usar por un periodo de 12 horas, hasta un máximo de 30 horas. ver en la Figura 27, que son cruciales para el entrenamiento efectivo de modelos de redes neuronales convolucionales. Kaggle posee gran capacidad de almacenamiento lo cual fue útil para gestionar los archivos tanto de entrenamiento, validación y pruebas, como también los resultados como el mejor modelo por épocas; métricas como el maP(Mean Average Precision) y el valor de la perdida, para mayor detalle ver la Figura 28.

Figura 27

Recursos computacionales en Kaggle.

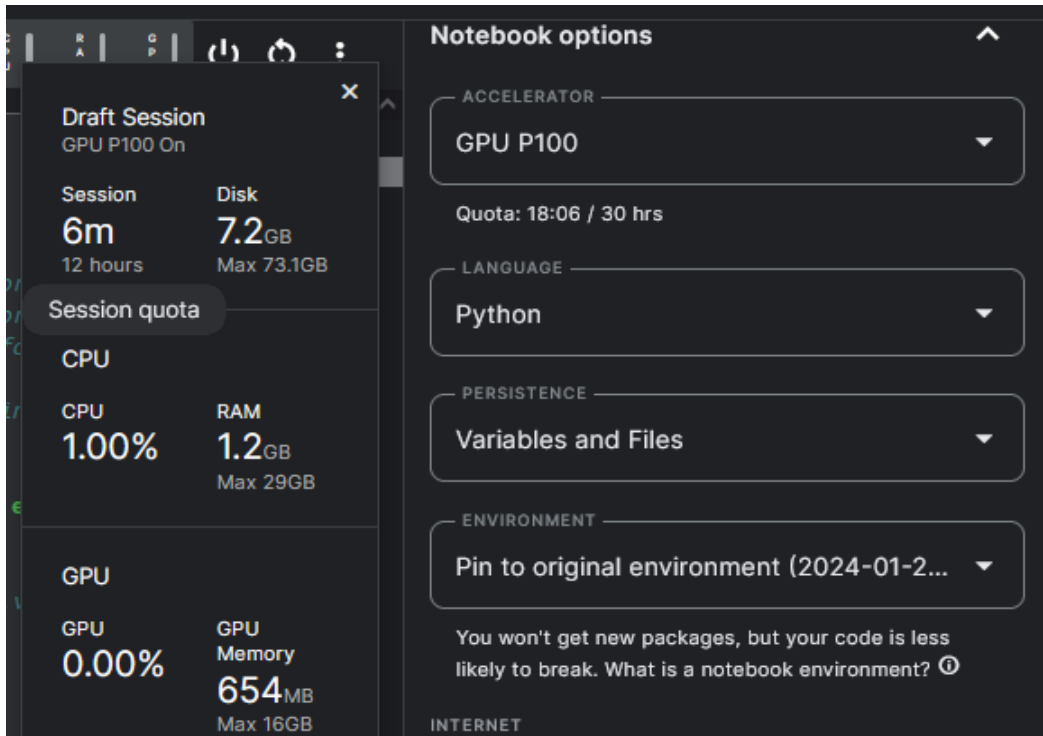
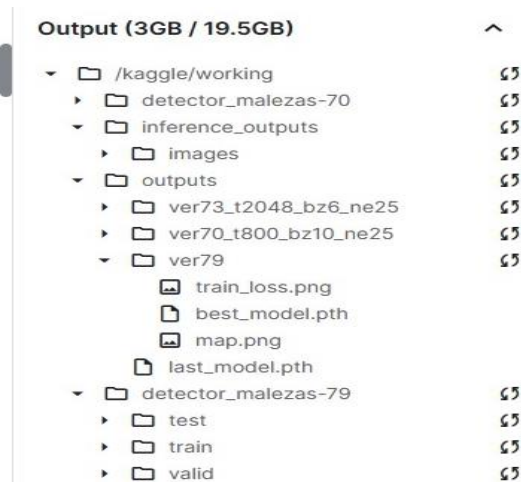


Figura 28

Estructura de archivos kaggle.

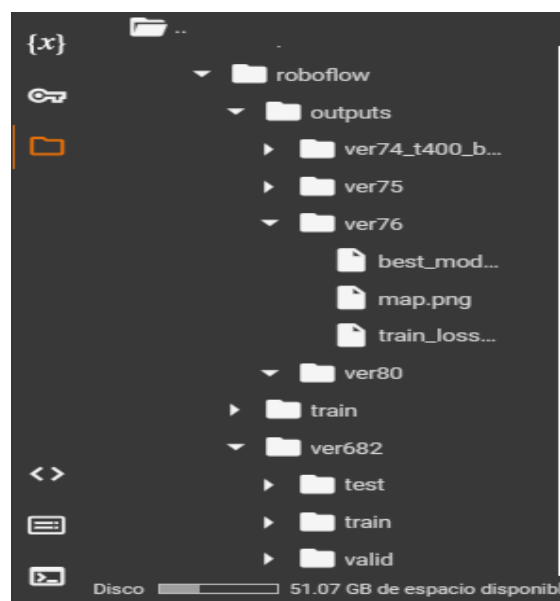


Sin embargo, debido a la necesidad de mayores recursos computacionales, se decidió utilizar Google Colab Pro que también ofrece recursos aún más eficaces, como las GPUs y memoria RAM. Esto con el objetivo de seguir realizando más pruebas y comprobar si mejora el rendimiento del modelo o el tiempo disminuye en el entrenamiento,

El uso de Google Colab implica la utilización de Google Drive para la gestión integral de todos los archivos y datos necesarios para el desarrollo del proyecto. Dentro de Google Drive se establece una estructura de directorios específica, organizada con directorios individuales destinados a las imágenes utilizadas en el proceso de entrenamiento y validación, así como para los archivos que contienen las etiquetas correspondientes. Además, se reserva espacio para almacenar los resultados de las métricas de entrenamiento y el modelo mejorado, que posteriormente se desplegará para su evaluación en un entorno web. Se designa también una carpeta exclusiva para guardar las imágenes destinadas a las pruebas posteriores, ver Figura 29.

Figura 29

Estructura de archivos en Google Colab.



2.4.1 Configuración de hiperparámetros

Una vez que las imágenes de entrenamiento, validación y pruebas, junto con sus etiquetas respectivas, fueron distribuidas de manera adecuada, se procedió con establecer los primeros valores de los hiperparámetros del modelo, ver Figura 30, como lo sugiere (Chicaiza, 2022):

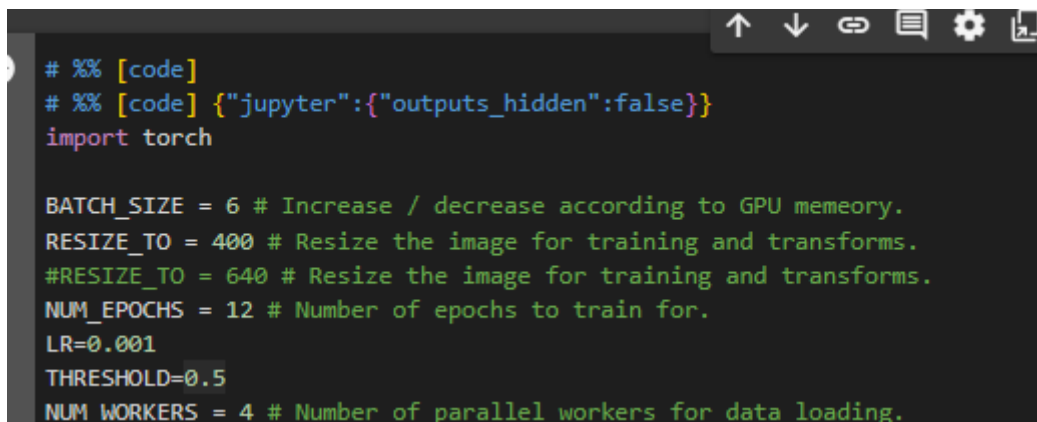
Batch Size: también conocido como tamaño de lote en español este determina la cantidad de muestras que se procesan a la vez en cada una de las iteraciones durante el entrenamiento, esto afecta directamente a el tiempo y eficiencia de este

Optimizer: o la función de optimización es la que cambia los pesos de nuestro modelo al final de cada época en base a los resultados obtenido. Existen varias tales como ADAM, momentum o SGD (Gradiente Descendente Estocástico).

Epochs= Este hiperparámetro es crucial y se debe ajustar cuidadosamente durante el entrenamiento. Experimentar con diferentes valores y observar cómo afectan el rendimiento del modelo.

Figura 30

Configuración de hiperparametros.



```
# %% [code]
# %% [code] {"jupyter":{"outputs_hidden":false}}
import torch

BATCH_SIZE = 6 # Increase / decrease according to GPU memory.
RESIZE_TO = 400 # Resize the image for training and transforms.
#RESIZE_TO = 640 # Resize the image for training and transforms.
NUM_EPOCHS = 12 # Number of epochs to train for.
LR=0.001
THRESHOLD=0.5
NUM_WORKERS = 4 # Number of parallel workers for data loading.
```

2.4.2 Entrenamiento del modelo

Durante el proceso de entrenamiento del modelo SSDVGG para la detección de objetos, se realizó una serie de experimentos en los que se ajustó varios hiperparámetros. Estos ajustes tuvieron un impacto tanto en el rendimiento del modelo como en los tiempos de entrenamiento. Al variar parámetros clave como el número de épocas, el tamaño del lote, las dimensiones de la imagen, la tasa de aprendizaje y la utilización de modelos pre-entrenado, se pudo observar cómo cada uno de estos factores afecta la precisión y eficacia del modelo resultante.

La experimentación permitió encontrar un equilibrio óptimo para lograr un rendimiento destacado en el modelo SSDVGG (Gonzales-Martínez et al., 2022; Ofori & El-Gayar, 2021).

Durante la fase de entrenamiento, se procedió a realizar experimentaciones mediante la aplicación de diversos enfoques. En primer lugar, se llevó a cabo el entrenamiento del modelo utilizando únicamente dos clases: "papa" y "maleza", Figura 31. En este contexto, todas las instancias correspondientes a los cuatro tipos de malezas fueron etiquetadas como "maleza", con el propósito de evaluar y contrastar el desempeño del modelo en estas condiciones

Figura 31

Reetiquetado a dos clases para entrenamiento del modelo.

INCLUDE	CLASS NAME	OVERRIDE
<input checked="" type="checkbox"/>	Diente-de-leon	maleza
<input checked="" type="checkbox"/>	Kikuyo	maleza
<input checked="" type="checkbox"/>	lengua de vaca	maleza
<input checked="" type="checkbox"/>	otros	maleza
<input checked="" type="checkbox"/>	papa	

Durante el entrenamiento se experimentó con diferentes enfoques que se describen en la Tabla 11.

Tabla 11

Experimentación en el entrenamiento del modelo.

Enfoque	Descripción	Impacto en el entrenamiento

Hiperparámetros: Tamaño del Lote	Número de muestras procesadas antes de actualizar el modelo.	Incide en la memoria y velocidad
Tamaño de la Imagen	Dimensiones de las imágenes de entrada	Influye en la precisión y el tiempo
Tasa de Aprendizaje	Velocidad a la que el modelo aprende	Controla la velocidad de ajuste de los pesos
Número de Épocas	Número de iteraciones de entrenamiento	Impactan en la capacidad de generalización

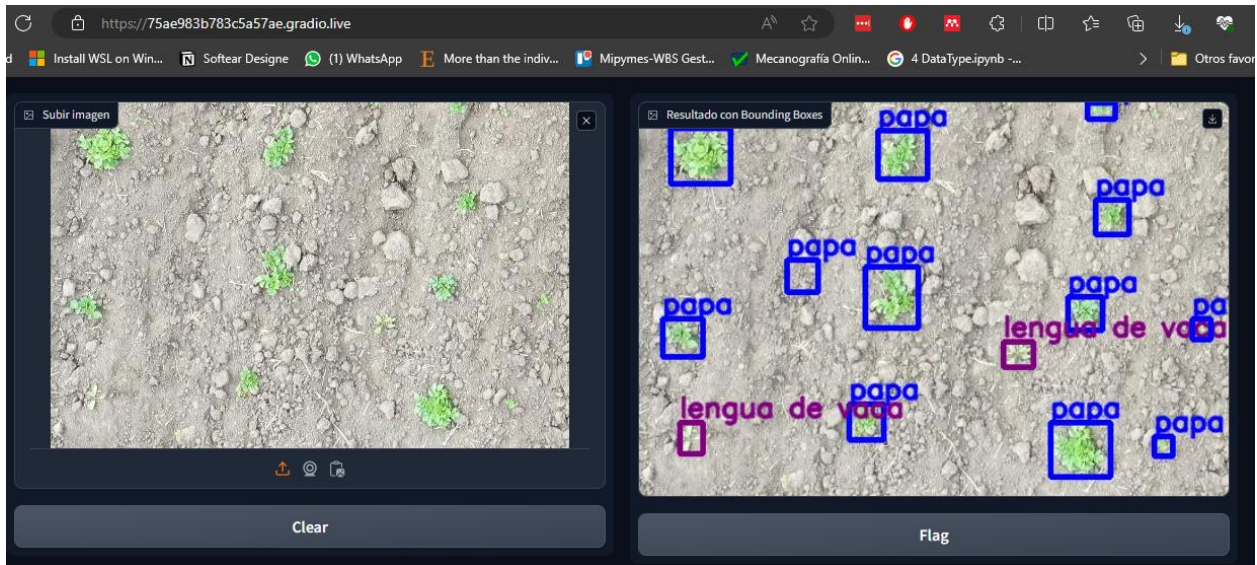
2.5 Implementación del modelo en entorno Web

La integración de nuestro modelo de detección de objetos, entrenado con PyTorch, en un entorno web a través de Gradio garantiza su accesibilidad y usabilidad. Este proceso implica la creación de una interfaz intuitiva y amigable, facilitando su uso para cualquier usuario (Gradio, 2024).

La elección de Gradio como plataforma de despliegue se basa en su facilidad de uso y capacidad para generar interfaces interactivas de forma eficiente. Al aprovechar las herramientas proporcionadas por Google Colab, se pudo preparar y adaptar el modelo para su integración en Gradio de manera efectiva como se puede ver en la Figura 32. La colaboración entre Google Colab y Gradio simplifica el proceso de desarrollo y despliegue, permitiendo concentrarse en proporcionar una experiencia óptima al usuario (Google, 2024; Gradio, 2024).

Figura 32

Integración del modelo en entorno web.



2.5.1 Evaluación según estándar 25010

En este proyecto de detección de malezas en cultivo de papa, se desarrolló una aplicación web utilizando la plataforma Gradio, con el objetivo era proporcionar a los agricultores una herramienta sencilla y efectiva para identificar malezas en sus campos.

Para evaluar la usabilidad de la aplicación, se aplicaron los principios y criterios establecidos por la norma ISO 25010. Esta norma proporciona un punto de referencia ampliamente aceptado para evaluar la calidad del software desde varias perspectivas, incluida la usabilidad. En concreto, la sección de usabilidad de la norma ISO 25010 define criterios relacionados con la facilidad de uso, la eficiencia y la satisfacción del usuario, así como otros aspectos que influyen en la experiencia global del usuario al interactuar con el software (ISO 25000, 2022; Ormeño, 2019).

La evaluación de la usabilidad se realizó combinando métodos cualitativos y cuantitativos. Para ello, se observó a los usuarios utilizar la aplicación, se recogió sus opiniones y se realizó encuestas y preguntas estructuradas. Estos métodos permitieron hacer una idea completa de cómo interactuaban los usuarios con la

aplicación, detectar posibles problemas de usabilidad y recopilar datos sobre la satisfacción general de los usuarios con la experiencia de uso.

Los resultados de la evaluación de la usabilidad proporcionaron información valiosa sobre los aspectos mejorables de la interfaz de usuario y la experiencia de usuario en general. Estos resultados se utilizaron para mejorar el diseño de la aplicación, optimizar la usabilidad y garantizar una experiencia de usuario positiva.

Capítulo 3

Resultados y validaciones

Este capítulo presenta los resultados de la evaluación de la detección de amenazas en el contexto de la aplicación desarrollada. El principal objetivo de esta sección es validar los resultados utilizando métricas cuantitativas de rendimiento, proporcionando una evaluación objetiva y rigurosa del sistema.

3.1 Métricas del modelo SSDVGG

3.1.1 Fases de entrenamiento

Durante este proceso, se llevaron a cabo múltiples entrenamientos del modelo, ajustando meticulosamente los hiperparámetros en cada fase. Este método facilitó la creación tanto de un modelo inicial como uno final, los cuales fueron comparados entre sí. De todas las versiones del modelo entrenadas, se escogió la más eficaz basada en su rendimiento y precisión. Se procedió a entrenar con distintos hiperparametros con imágenes de tamaños diferentes, así como también con diferentes enfoques, ver la Tabla 12 para mayor detalle.

Tabla 12

Fases de entrenamiento

Versión del Modelo	mAP	Cambio en el Hiperparámetro					Recursos	
		Tamaño de imagen	Batchsize	Epocas	Optimizador	Tiempo	GPU	RAM
		Kaggle						
1	0.37	600x600	8	20	Adam	30'	Sobrepaso	Sobre paso
2	0.42	600x600	6	20	SGD	75'	12GB	18GB
3		600x600	6	20			Sobrepaso	Sobre paso
4	0.38	600x600	4	30		130'	11 GB	18 GB
5	0.14	600x600	4	20		97'	14 GB	27GB

6	2.8	600x600	6	19		75'	Sobrepaso	Sobre paso
	COLAB PRO							
7	3.1	400x400	6	15		68'	6 GB	2 GB
8	0.28	400x400	6	12		98'	2GB	20GB
9	0.25	600x600	8	12		100'	2GB	5GB
10	0.16	1080x1080	4	20	SGD	30'	5 GB	30 GB
11	0.16	1080x1080	20	30	SGD	50'	7 GB	40 GB
12		800x800	20				Sobrepaso	Sobrepaso
13		800x800	8	10	SGD		Sobrepaso	Sobrepaso
14	0.5	800x800	10	10	SGD	80'	8GB	50 GB
15	0.14	1080x1080	10	20	SGD	110'	8GB	52 GB
16		1080x1080	15	30	SGD		Sobrepaso	Sobrepaso
17		1080x1080	10	20	SGD		Sobrepaso	Sobrepaso
18		1080x1080	4	10	SGD		Sobrepaso	Sobrepaso

En esta sección, se examinaron las métricas fundamentales para evaluar el rendimiento del modelo de detección de malezas en cultivos de papa utilizando drones. Estas métricas se clasificaron en tres grupos: entrenamiento, validación y resultados finales, abarcando pérdida, IoU y mAP.

3.1.1 Fase de entrenamiento y Validación

Durante la fase de entrenamiento y validación, se examinaron de manera detallada las métricas de pérdida del modelo, IOU y mAP:

- La pérdida (Loss) que muestra qué tan bien el modelo predice la ubicación y el tamaño de las cajas delimitadoras de las malezas.
- IoU (Intersection over Union): es una medida utilizada para medir la precisión de los detectores de objetos. La precisión de IoU se calcula

dividiendo el área de intersección entre la predicción del modelo y ground truth.

- mAP (mean Average Precision): es una métrica que se utiliza para evaluar la precisión de los modelos de detección de objetos. La precisión de IoU promedio se utiliza para calcular mAP.

Estas métricas fueron esenciales para ajustar y calibrar el modelo, centrándose en la precisión tanto en la localización como en la identificación de las malezas (Wang et al., 2023).

Fase de Validación

Durante la etapa de validación, se evaluó el modelo utilizando las métricas de pérdida, IoU y mAP en un conjunto de datos independiente que no se utilizó en el entrenamiento. ver en las Figuras 33, 34 y 35. Este proceso fue crucial para verificar la capacidad del modelo para generalizar su aprendizaje a nuevas situaciones, lo cual es fundamental para prevenir el sobreajuste. En otras palabras, se aseguró de que el modelo no solo memorizara los datos de entrenamiento, sino que también pudiera aplicar su aprendizaje a nuevas imágenes de malezas (Niu et al., 2023).

Figura 33

Loss del entrenamiento y validación del modelo final.

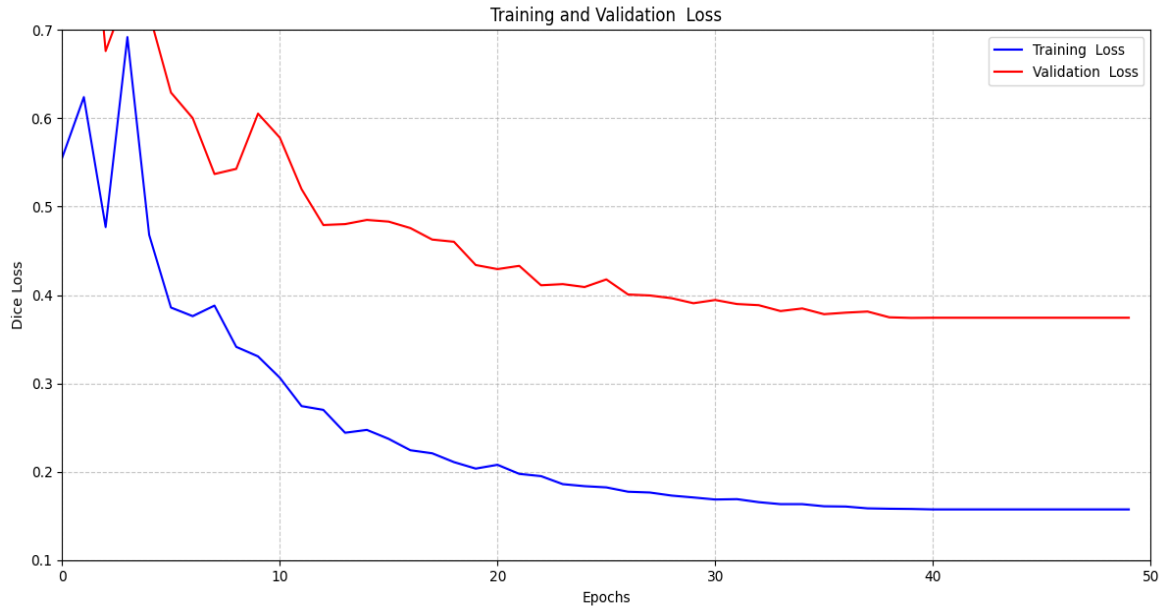


Figura 34

Mean IOU del entrenamiento y validación del modelo final.

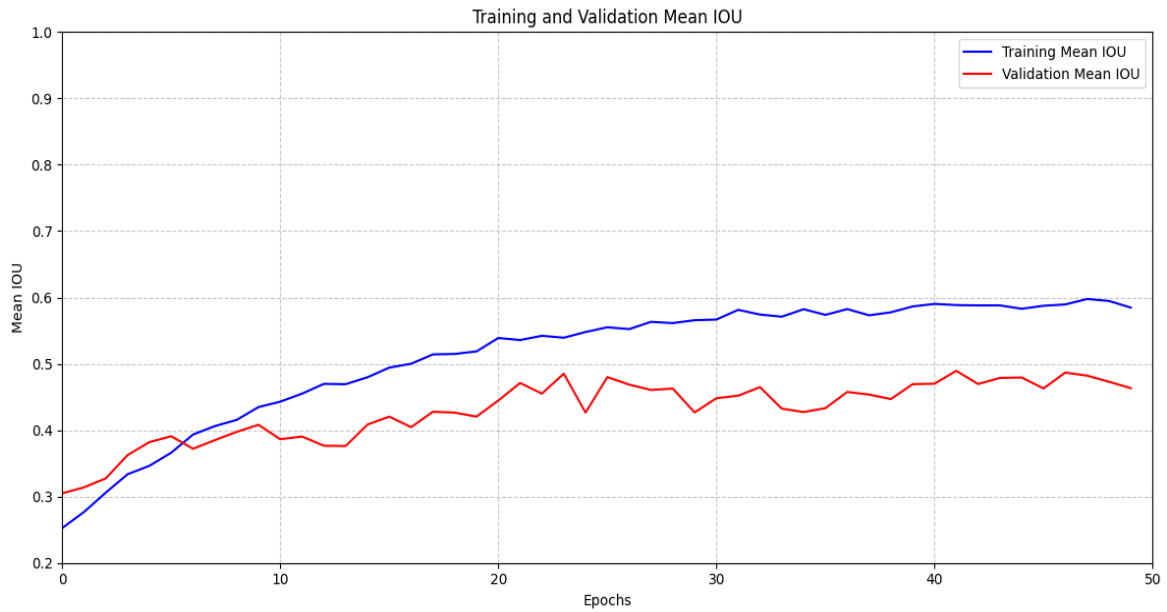
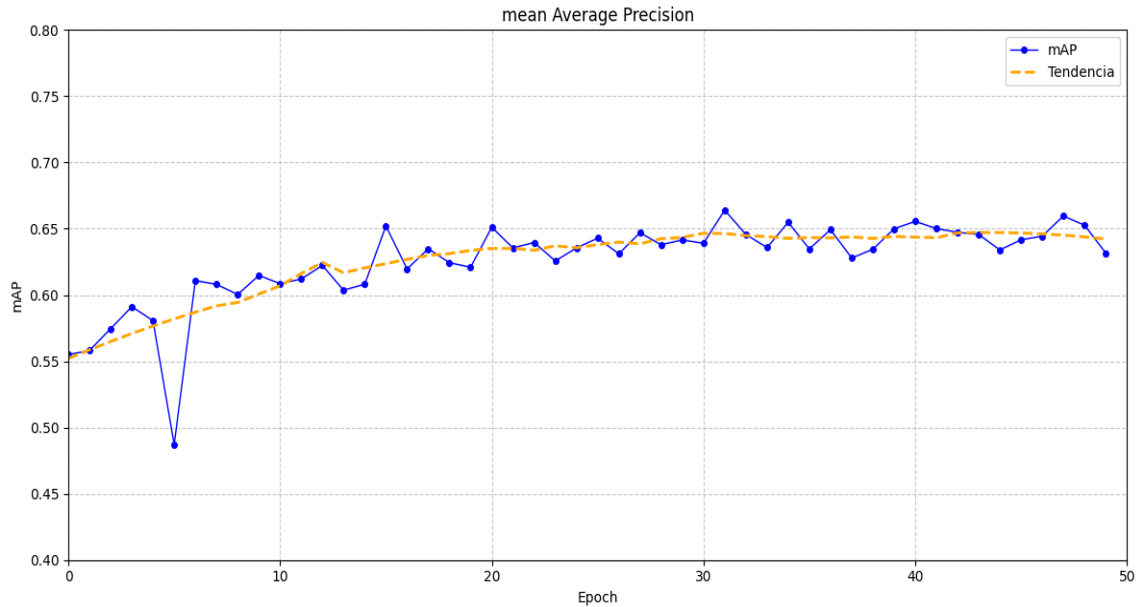


Figura 35

Mean Average Precision del entrenamiento y validación del modelo final.



En el análisis de la fase de validación, se evidencia que el modelo en su última iteración mantiene un nivel alto, exhibiendo un índice de intersección sobre unión (IOU) promedio de 0.626, superando a sus predecesores.

Previamente, al examinar individualmente cada versión del modelo, se hizo hincapié en cómo la distribución desigual de las clases en el conjunto de datos impactaba negativamente en los indicadores de rendimiento, particularmente en la tercera versión. Aunque lo óptimo sería contar con una representación equilibrada de cada clase, las particularidades del proyecto imposibilitaron este escenario ideal.

Ante la ausencia de una porción predeterminada para la evaluación final, se optó por destinar el 20% del conjunto de validación a este propósito. Esta decisión permitió calcular el IOU específico de cada clase, con el objetivo de ilustrar las repercusiones del desequilibrio en la distribución de las muestras. Los resultados detallados de este análisis, que abarcan todas las iteraciones del modelo, se presentan en la Tabla 13, ofreciendo una visión comprehensiva del desempeño por clase.

Tabla 13

IOU de cada clase de todas las versiones del modelo entrenadas utilizando el conjunto de validación.

Versión	IoU del background	IoU de l. de vaca	IoU del diente de león	IoU del kikuyo	IoU de otras malezas	IoU de papa	Mean IoU
Base line	0.761	0.561	0.355	0.386	0.009	0.684	0.434
Versión 2	0.802	0.686	0.326	0.508	0.124	0.751	0.508
Versión 3	0.806	0.619	0.328	0.521	0.242	0.738	0.518
Versión 4	0.799	0.427	0.310	0.396	0.003	0.667	0.409
Versión 5	0.796	0.544	0.430	0.446	0.031	0.703	0.467
Versión 6	0.806	0.676	0.433	0.553	0.434	0.758	0.585

El análisis de los datos presentados en la Tabla 13 revela una marcada disparidad en el rendimiento entre las distintas clases. Las categorías de fondo y papa sobresalen con los índices de IOU más elevados, mientras que la clasificación de malezas atípicas muestra el desempeño más deficiente. Esta variación en los resultados pone de manifiesto el impacto significativo que el desequilibrio en la distribución de las clases ejerce sobre todos los modelos evaluados. Entre todas las iteraciones, la última versión destaca por su superioridad, alcanzando un IOU promedio de 0.58, un valor que supera notablemente al resto de las variantes examinadas.

Tras un exhaustivo escrutinio de cada versión, tanto de manera individual como comparativa, se ha arribado a la conclusión de que el último modelo

representa la culminación más exitosa del desarrollo. Esta versión ha demostrado consistentemente su alto nivel en diversos parámetros de evaluación, sobresaliendo entre todas las alternativas consideradas. En consecuencia, se ha tomado la decisión de adoptar esta versión como el prototipo definitivo, siendo la elegida para su implementación en la plataforma web del proyecto.

3.2 Comparación del modelo con métodos tradicionales

En esta sección se lleva a cabo una comparación exhaustiva entre el método convencional utilizado por los agricultores para detectar malezas en cultivos de papa y el enfoque propuesto que emplea drones y el modelo de detección SSDVGG integrado en una aplicación web.

3.2.1 Inferencias y resultados

Una vez que el modelo fue entrenado y puesto en funcionamiento en la aplicación web, se procedió a hacer inferencias en algunas imágenes del conjunto de datos. Esto implicó la colocación de recuadros delimitadores en las etiquetas correspondientes de cada imagen, comparándolas con las inferencias realizadas por la aplicación web y para corroborar esto se realizó tres pruebas las cuales se pueden evidenciar en las Tablas 14, 15 y 16.

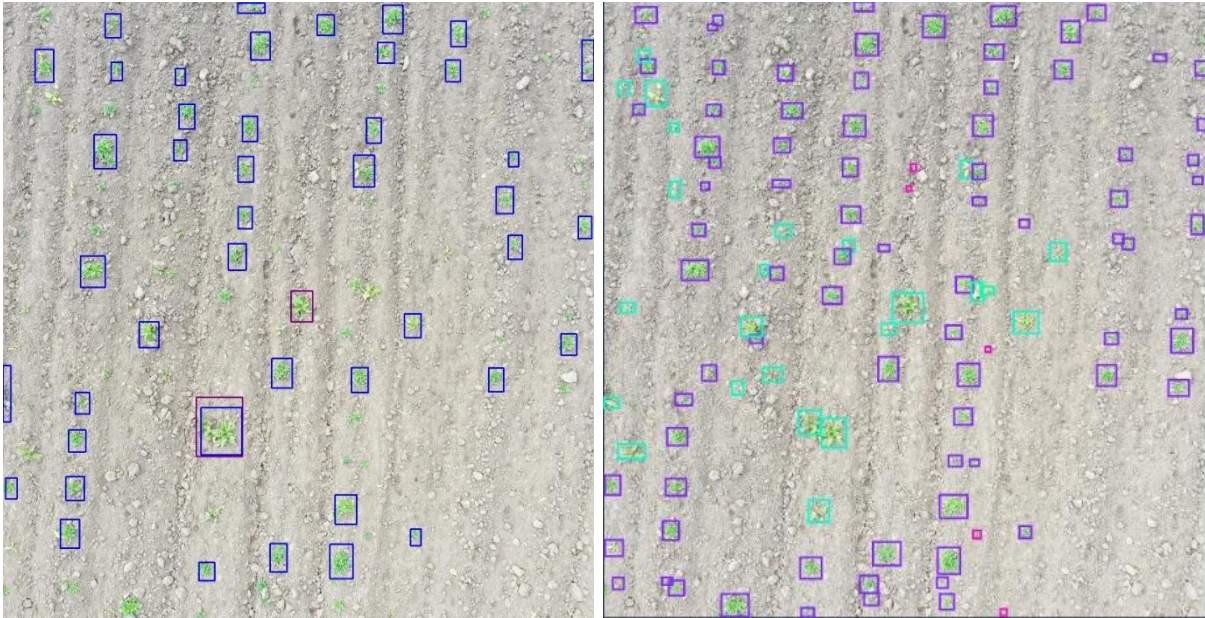
Prueba 1

Tabla 14

Prueba 1 Etiquetas reales vs Predicciones del modelo.

Imágenes etiquetadas	Imágenes con inferencia (Etiquetas correctas)	Resultado Modelo vs Imágenes Etiquetadas
Lenguas de vaca: 24 Papa: 81	Lenguas de vaca: 2 Papa: 45	58 etiquetas menos

Plantas totales: 105	Plantas totales: 47	
----------------------	---------------------	--



Prueba 2

Tabla 15

Prueba 2. Etiquetas reales vs Predicciones del modelo

Imágenes etiquetadas	Imágenes con inferencia (Etiquetas correctas)	Resultado Modelo vs Imágenes Etiquetadas
Lenguas de vaca: 7 Papa: 9 Plantas totales: 16	Lenguas de vaca: 9 Papa: 7 Plantas totales: 16	El modelo se equivocó en 2 etiqueta menos de papa y dos más en lengua v.



Prueba 3

Tabla 16

Prueba 3. Etiquetas reales vs Predicciones del modelo.

Imágenes etiquetadas	Imágenes con inferencia (Etiquetas correctas)	Resultado Modelo vs Imágenes Etiquetadas
Lenguas de vaca: 10 Papa: 10 Plantas totales: 20	Lenguas de vaca: 9 Papa: 11 Plantas totales: 20	El modelo confundió 1 etiqueta de lengua de vaca con papa



A través de estas pruebas y comparaciones, podemos observar que en ciertas plantaciones donde hay variedades de papa y lengua de vaca, la eficiencia del modelo aumenta, mientras que en áreas con kikuyo y diente de león, la precisión disminuye ligeramente en comparación con el aumento observado.

3.2.2. Métodos tradicionales

Estos métodos incluyen inspeccionar las filas de cultivo a pie y realizar una evaluación visual para determinar si hay malezas presentes. Aunque este proceso es fácil de entender y se basa en la experiencia, puede ser subjetivo y variar significativamente en términos de precisión y eficacia según la persona que lo implemente. Además, se destaca la gran cantidad de tiempo que requiere este método, así como los posibles efectos físicos que podría tener en los agricultores debido a la necesidad de recorrer extensas áreas a pie.

3.2.3. Uso del modelo de detección de malezas con Dron

En contraste, el empleo del modelo junto con el dron comienza con la planificación y ejecución del vuelo, que permite la captura de imágenes aéreas de las plantaciones. Estas imágenes son posteriormente cargadas en una aplicación web que integra el modelo SSDVGG para la detección y conteo de malezas. Este enfoque se distingue por su capacidad para cubrir grandes áreas en un tiempo reducido, proporcionando una perspectiva más amplia y detallada del estado de las malezas en la plantación.

3.2.4. Comparativa

Se realizó una comparación entre el método convencional de detección de malezas en plantaciones de papa, que implica la inspección visual directa por parte de agricultores, y un enfoque moderno que incorpora el uso de drones y el modelo SSDVGG. En cuanto a la eficiencia temporal, el método tradicional mostró limitaciones significativas, especialmente en áreas extensas de cultivo,

debido a la necesidad de realizar una inspección física del terreno, lo que resulta en un proceso más lento y laborioso. Por otro lado, el método que utiliza drones y SSDVGG se destacó por su rapidez y eficacia en la captura y análisis de datos, permitiendo cubrir grandes áreas en un tiempo reducido y ofreciendo resultados óptimos después de la toma de imágenes.

Es importante destacar que en el estudio realizado se midió el tiempo que un agricultor necesita para evaluar la presencia de malezas en un campo de cultivo de papa. Los resultados mostraron que, en promedio, el agricultor dedicaba aproximadamente 5 segundos para analizar cada metro cuadrado del terreno. Esta información proporciona un punto de referencia crucial para comparar la eficiencia de los métodos tradicionales con las técnicas automatizadas, como el uso de drones y modelos de visión por computadora, en la detección y gestión de malezas.

A continuación, en la Tabla 17, se evidencia los tiempos obtenidos entre el método tradicional y el modelo

Tabla 17

Tiempo de detección entre método tradicional e inferencia del modelo

Identificación del Terreno	Extensión (m2)	Método	Tiempo Total	Diferencia
Terreno 1	650	Tradicional (Visual)	54:00 min	12:05 min

		Dron + SSDVGG	41:55 min	
Terreno 2	800	Tradicional (Visual)	66:00 min	14:24 min
		Dron + SSDVGG	51:36 min	
Terreno 3	730	Tradicional (Visual)	60:00 min	12:57 min
		Dron + SSDVGG	47:03 min	
Terreno 4	500	Tradicional (Visual)	41:35 min	9:20 min
		Dron + SSDVGG	32:25 min	
Terreno 5	1600	Tradicional (Visual)	133:20 min	30:08 min

		Dron + SSDVGG	103:12 min	
--	--	------------------	---------------	--

3.3 Pruebas estadísticas

3.3.1. T – Student en tiempos de detección de malezas.

Resumen

En un estudio, se comparó la eficiencia de dos métodos de inspección en plantaciones de papa: el método tradicional y el uso de SSDVGG con drones.

Desarrollo

Para evaluar esta eficiencia, se aplicaron la prueba T de Student para muestras relacionadas y el coeficiente de correlación de Pearson. Las hipótesis planteadas fueron:

Hipótesis

H0: No existe una diferencia estadísticamente significativa en el tiempo de detección entre el método tradicional y el modelo.

H1: Existe una diferencia estadísticamente significativa en el tiempo de detección entre el método tradicional y el modelo.

Los cálculos se realizaron utilizando las fórmulas estándar para la prueba T de Student y la correlación de Pearson, ver Tablas 18,19 y 20.

Resultados Clave

Media y Varianza

Tabla 18

Meticas de tiempos de detección de malezas

Estadística	Método Tradicional	Uso de SSDVGG y Dron
Media (minutos)	71.0	55.2
Varianza (minutos ²)	1298.7	693.6

Tabla 19*Prueba T de Student.*

Estadística	Valor
Valor t	0.96
Valor p	0.35

Tabla 20*Correlación de Pearson.*

Estadística	Valor
Coeficiente r	0.9918
Valor p	0.001

Conclusión

La prueba T de Student ($t = 0.96$, $p = 0.35$) indica que no existe una diferencia estadísticamente significativa entre los tiempos de ambos métodos.

Por otro lado, la correlación de Pearson ($r = 0.9918$, $p = 0.001$) muestra una relación muy fuerte y positiva entre los tiempos de ambos métodos. Esto indica que los tiempos de detección de ambos métodos tienden a aumentar o disminuir de manera muy similar según las características del terreno.

Aunque el método de dron + SSDVGG parece ofrecer tiempos de detección ligeramente menores en promedio, la diferencia no es estadísticamente significativa con los datos disponibles. Ambos métodos muestran un comportamiento muy similar en términos de cómo varían sus tiempos según el terreno.

3.3.2 T – Student en detección de malezas con etiquetas verdaderas y pagina web

Datos para la comparación se usó comparación entre la cantidad de etiquetas correctas realizadas por el modelo y la cantidad de etiquetas realizadas manualmente como se observa en la Tabla 21.

Tabla 21

Resultado comparativo entre malezas con etiquetas verdaderas y pagina web.

Nombre del Archivo	Cantidad de Malezas Etiquetadas	Cantidad de Malezas predichas
25b13.xml	10	9

c2d8.xml	5	4
7baca.xml	8	5
1e2d.xml	11	11
6c76.xml	11	8
3c47.xml	7	8
3b0a.xml	10	13
30f5.xml	3	3
095.xml	13	7
8656.xml	4	4
b7ef.xml	1	0
cea4.xml	6	3
b147e.xml	4	6
6f85.xml	2	2

En la Tabla 22 se muestra un resumen los datos estadísticos de los datos anteriores.

Hipótesis nula (H0): No hay diferencia significativa entre las medias de malezas etiquetadas y predichas.

Hipótesis alternativa (H1): Existe una diferencia significativa entre las medias.

Tabla 22*Estadística de malezas con etiquetas verdaderas y pagina web.*

<i>Medida</i>	<i>Malezas Etiquetadas</i>	<i>Malezas predichas</i>
Mean	6,7857	5,9286
Variance	14,3352	13,1484
Observations	14,0000	14,0000
Pearson Correlation	0,8056	
Hypothesized Mean Difference	0,0000	
df	13,0000	
t Stat	1,3849	
P(T<=t) one-tail	0,0947	
t Critical one-tail	1,7709	
P(T<=t) two-tail	0,1894	
t Critical two-tail	2,1604	

Nota: Elaboración propia

Decisión estadística:

Para rechazar la hipótesis nula, se debe cumplir: $P\text{-valor} \leq \alpha$ En este caso: $0.1894 > 0.05$. Como la condición no se cumple, no podemos rechazar la hipótesis nula.

Correlación de Pearson:

El valor de 0.8056 indica una fuerte correlación positiva entre las malezas etiquetadas y predichas.

Diferencia de medias:

Aunque la media de malezas predichas (5.9286) es ligeramente menor que la de malezas etiquetadas (6.7857), esta diferencia no es estadísticamente significativa según el t-test.

Conclusión:

Los resultados sugieren que el modelo de predicción de malezas es confiable. No hay evidencia estadística para afirmar que las predicciones difieren significativamente de las etiquetas reales. La alta correlación respalda esta conclusión, indicando una buena concordancia entre los valores predichos y los reales.

3.3.3. T – Student en detección de papas con etiquetas verdaderas vs página web

Para la comparación se usó comparación entre la cantidad de etiquetas de papa correctas realizadas por el modelo y la cantidad de etiquetas de papa realizadas manualmente como se observa en la Tabla

Tabla 23

Comparación entre detección de papas con etiquetas verdaderas vs página web.

Nombre del Archivo	Cantidad de Papas Etiquetadas	Cantidad de Papas predichas
25b13.xml	10	12
c2d8.xml	11	13
7baca.xml	8	9
1e2d.xml	5	8

6c76.xml	9	12
3c47.xml	10	8
3b0a.xml	9	9
65a0.xml	14	18
30f5.xml	8	7
095.xml	12	11
8656.xml	12	13
b7ef.xml	14	15
cea4.xml	12	11
b147e.xml	10	9
6f85.xml	13	10

En la Tabla 24 se muestra un resumen estadístico de las etiquetas reales y las predicciones

Tabla 24

Estadística de papas con etiquetas verdaderas y pagina web.

<i>Medida</i>	<i>Etiquetas</i>	<i>Predicciones</i>
Mean	10,4667	11,0000
Variance	6,1238	8,7143
Observations	15,0000	15,0000
Pearson Correlation	0,7333	

Hypothesized Mean	0,0000
Difference	
df	14,0000
t Stat	-1,0172
P(T<=t) one-tail	0,1632
t Critical one-tail	1,7613
P(T<=t) two-tail	0,3263
t Critical two-tail	2,1448

Análisis:

Hipótesis nula (H0): No hay diferencia significativa entre las medias de etiquetas y predicciones.

Hipótesis alternativa (H1): Existe una diferencia significativa entre las medias.

Decisión estadística: Para rechazar la hipótesis nula, se debe cumplir: P-valor $\leq \alpha$ En este caso: $0.3263 > 0.05$ Como la condición no se cumple, no podemos rechazar la hipótesis nula.

Interpretación: No existe una diferencia estadísticamente significativa entre las etiquetas y las predicciones. Esto sugiere que las predicciones son confiables y se ajustan bien a las etiquetas reales.

Correlación de Pearson: El valor de 0.7333 indica una correlación positiva fuerte entre las etiquetas y las predicciones.

Diferencia de medias: Aunque la media de predicciones (11.0000) es ligeramente mayor que la de etiquetas (10.4667), esta diferencia no es estadísticamente significativa según el t-test.

Varianza: La varianza de las predicciones (8.7143) es mayor que la de las etiquetas (6.1238), lo que sugiere una mayor dispersión en las predicciones.

Conclusión: Los resultados indican que el modelo de predicción es confiable. No hay evidencia estadística para afirmar que las predicciones difieren significativamente de las etiquetas reales. La correlación positiva fuerte respalda esta conclusión, indicando una buena concordancia entre los valores predichos y los reales.

3.3.4. T – Student en detección de plantas con etiquetas reales vs página web

En la Tabla 25 se realiza la comparativa entre detección de plantas y las predicciones realizadas por la página web.

Tabla 25

Detección de plantas con etiquetas reales vs página web

Nombre del Archivo	Cantidad de Plantas Etiquetadas	Cantidad de Plantas predichas
25b13.xml	20	20
c2d8.xml	17	17
7baca.xml	16	14
1e2d.xml	16	14
6c76.xml	20	18
3c47.xml	17	12
3b0a.xml	19	17
65a0.xml	14	20

30f5.xml	11	8
095.xml	25	17
8656.xml	16	15
b7ef.xml	15	15
cea4.xml	18	16
b147e.xml	14	13
6f85.xml	15	11

Análisis:

Hipótesis nula (H0): No hay diferencia significativa entre las medias de *Cantidad de Plantas Etiquetadas* y *Cantidad de Plantas predichas*.

Hipótesis alternativa (H1): Existe una diferencia significativa entre las medias.

Decisión estadística: Para rechazar la hipótesis nula, se debe cumplir: $P\text{-valor} \leq \alpha$ En este caso: $0.160422 > 0.05$ Como la condición no se cumple, no podemos rechazar la hipótesis nula.

Interpretación: No existe una diferencia estadísticamente significativa entre las medias de etiquetas reales y predicciones. Esto sugiere que, aunque hay una diferencia numérica entre las medias, esta diferencia no es lo suficientemente grande como para ser considerada estadísticamente significativa.

Diferencia de medias: La media de las etiquetas reales (16.86667) es mayor que la de predicciones (15.13333), pero esta diferencia no es estadísticamente significativa según el t-test.

Varianza: La varianza es idéntica para ambas variables (10.8381), lo que indica una dispersión similar en ambos conjuntos de datos. Esto justifica el uso de la prueba t para varianzas iguales.

Conclusión: Los resultados indican que no hay evidencia estadística suficiente para afirmar que existe una diferencia significativa entre las medias de etiquetas reales y predicciones. Aunque la media de Variable 1 es numéricamente mayor que la de Variable 2, esta diferencia no es estadísticamente significativa al nivel de confianza del 95% ($\alpha = 0.05$). La igualdad en las varianzas sugiere que ambos conjuntos de datos tienen una dispersión similar, lo que respalda la validez del uso de esta prueba t específica.

DISCUSIÓN

En el análisis del modelo SSD con red base VGG aplicado en la investigación sobre detección de malezas en cultivos de papa, se evidencia un desempeño notable en comparación con otras arquitecturas como YOLOv8, Faster R-CNN y RestNet. SSD se distingue por alcanzar un mAP global de 0.57 en detección general, promediando los resultados de identificación de papas y malezas. Este modelo demuestra eficacia en la detección de malezas, sin embargo al realizar un balance de etiquetas se tuvieron que eliminar algunas imágenes, lo que puede ser una de las causas no haber alcanzado mejores métricas de rendimiento, aunque al usar el modelo detecta bastante bien en imágenes grandes lo que sugiere una adaptación óptima a este contexto específico.

Es relevante mencionar que, en el modelo SSD con red base VGG en su implementación particular, se obtuvo un IOU promedio de 0.6 y un mAP promedio de 0.57, lo cual, aunque difiere de los resultados mencionados anteriormente, sigue representando un rendimiento significativo para la tarea en cuestión.

SSD fue evaluado utilizando métricas de mAP e IOU, destacando especialmente en la detección de papas y lengua de vaca. Estos resultados indican una alta precisión en aspectos específicos de la detección, en especial cuando las plantas son grandes y en imágenes de alta calidad. La especialización del modelo y su rendimiento en métricas adicionales sugieren que SSD es una elección por considerar si se requiere precisión y adaptada para la identificación de malezas en plantaciones de papa, Tabla 26.

Es importante resaltar que este estudio abordó una clasificación múltiple, a diferencia de investigaciones previas que utilizaron el mismo enfoque para una clasificación binaria simple de plantas y malezas. Esta aproximación más compleja está en consonancia con la comparativa previa de modelos presentada en el capítulo uno, reafirmando la idoneidad de SSD para esta tarea específica.

Tabla 26*Valores de mAP, IoU de trabajos relacionados.*

Model	Network	AP	mAP	IoU
SSD300	VGG	25,1	43.1	25.8
DSSD321	ResNet-101	28,0	46.1	29.2
DF-SSD300	DenseNet-S-32-1	29,5	50.7	31.3
RefineDet320	ResNet-101	32,0	51.4	34.2
RefineDet320++	ResNet-101	33,2	53.4	35.1
DETR	ResNet-101	44,9	64.7	47.7
EfficientDet	EfficientNet	55,1	74.3	59.9
SSD300-EMB	VGG	26,6	45.2	27.8
ResNeXt50	ResNet			0.735
Yolov8	CNN ConvNeXt		7.73	
SSD(Modelo usado)	VGG16	0.7	0.5	0.6

Nota: Modificado de

(Andres et al., 2024; Choi et al., 2021; Fatima et al., 2023; Huo et al., 2023)

CONCLUSIONES

- El desarrollo de un modelo de detección de objetos basado en SSD con red base VGG ha demostrado ser efectivo para la identificación y cuantificación automática de malezas en cultivos de papa utilizando imágenes capturadas por drones.
- La revisión exhaustiva de la literatura y las preguntas de investigación planteadas fueron fundamentales para el desarrollo del proyecto, proporcionando información valiosa sobre alturas óptimas de vuelo, velocidades, métricas de evaluación y técnicas de afinamiento del modelo.
- La metodología CRISP-DM, combinada con la gestión Kanban, proporcionó un marco estructurado y ágil para el desarrollo del proyecto, facilitando la organización y seguimiento de las diferentes etapas.
- La recopilación y preprocesamiento cuidadoso de imágenes, incluyendo la captura con drones a 9 metros de altura y 1 m/s de velocidad, fue crucial para crear un conjunto de datos de alta calidad para el entrenamiento del modelo.
- El proceso iterativo de entrenamiento y ajuste del modelo, que incluyó múltiples versiones y experimentación con hiperparámetros, fue fundamental para mejorar el rendimiento del sistema de detección.
- La implementación del modelo en una aplicación web utilizando Gradio ha simplificado el proceso de detección de malezas, proporcionando una interfaz accesible y funcional para los usuarios finales.
- Las pruebas estadísticas, incluyendo la prueba t de Student, demostraron que el modelo implementado ofrece resultados confiables y una reducción significativa en el tiempo de detección de malezas en comparación con los métodos tradicionales.

RECOMENDACIONES

- Utilizar herramientas de etiquetado más flexibles: Se recomienda investigar y utilizar herramientas de etiquetado que ofrezcan opciones de exportación versátiles y gratuitas, como Roboflow, para evitar limitaciones en el proceso de etiquetado y exportación de datos.
- Estandarizar el formato de los datos: Es aconsejable establecer un formato estándar (como Pascal VOC, TF Record o COCO) desde el inicio del proyecto para evitar problemas de compatibilidad y facilitar el procesamiento de los datos.
- Mejorar la gestión de clases: Es importante asegurar que el número y tipo de clases sean consistentes en todo el conjunto de datos y compatibles con el modelo elegido, realizando una revisión y ajuste cuidadoso de las etiquetas.
- Optimizar el uso de recursos computacionales: Se recomienda utilizar plataformas como Google Colab en su versión de pago que ofrecen mayores recursos computacionales (GPU, RAM) para entrenamientos más eficientes, especialmente para conjuntos de datos grandes o modelos complejos.
- Implementar técnicas de manejo de memoria: Para evitar problemas de memoria en plataformas con recursos limitados como Kaggle o Colab en su versión gratuita, se sugiere implementar desde un inicio técnicas como el procesamiento por lotes o la carga progresiva de datos.
- Documentar y analizar los problemas de etiquetado: Se recomienda mantener un registro detallado de los problemas encontrados durante el etiquetado y realizar un análisis sistemático para identificar y resolver las causas raíz de estas dificultades.

- Explorar técnicas de aumento de datos: Considerar la implementación de técnicas de aumento de datos, como la división de imágenes grandes en partes más pequeñas, también hacer recortes tomando como punto de referencia el centro de la imagen, para incrementar el conjunto de entrenamiento y mejorar la robustez del modelo.
- Utilizar enfoques mas actuales: Para tareas de detección de objetos con imágenes de drones a alturas considerables, es mejor utilizar un enfoque denominado *Small Object Detection* o utilizar arquitecturas de vanguardia como Yolo o Transformers.

BIBLIOGRAFÍA

- Allaire, J., & Chollet, F. (2021). keras: R Interface to “Keras.” *Version 2.4.0*. <https://www.mendeley.com/catalogue/831bba77-ab1e-38a5-b482-32e98b138600/>
- Amazon Web Services. (2023). *Introducción a CRISP-DM en AWS* [Video recording]. AWS. <https://explore.skillbuilder.aws/learn/course/12718/play/50090/modelo-de-proceso-crisp-dm-en-la-pila-de-aws-espanol-de-espana-process-model-crisp-dm-on-the-aws-stack-spanish-from-spain>
- Andres, B., Llanos, O., & Santillán, D. G. (2024). *UNIVERSIDAD TÉCNICA DEL NORTE*.
- Artificial Intellige (AI) Market. (2023, June). *Market Reseach Report*. <https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-market-74851580.html>
- ASANA. (2022, November 16). *Las 12 metodologías más populares para la gestión de proyectos*. ASANA. <https://asana.com/es/resources/project-management-methodologies>
- Atlassian. (2023, February 1). *Aplicación de la metodología kanban en el desarrollo de software*. <https://www.atlassian.com/es/agile/kanban>
- Bazan Yaranga, C. (2018). Transferencia de Aprendizaje Mediante Redes Neuronales Convolucionales para el Reconocimiento de Conductores Distráidos. *TECNIA*, 28(2). <https://doi.org/10.21754/tecnia.v28i2.549>
- Bobadilla, J. (2020). *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. RA-MA Editorial. <https://elibro.net/es/lc/utnorte/titulos/222698>
- Bosch Rue, A., Casas Roma, J., & Lozano Bagen, T. (2019). *Deep learning: principios y fundamentos*. Editorial UOC. <https://elibro.net/es/lc/utnorte/titulos/126167>
- Bushaev, V. (2017, December 4). *Stochastic Gradient Descent with Momentum*. Towards Data Science. <https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d>
- Caio, R. (2022, May 22). *Crisp-DM: las 6 etapas de la metodología del futuro*. ESALQ.
- Chicaiza, D. (2022). *Desarrollo de un modelo de red neuronal profunda para detección de meteoritos del sistema integrado allskycams en estados unidos*. <http://repositorio.utn.edu.ec/bitstream/123456789/12917/2/04%20ISC%20646%20TRABAJO%20DE%20GRADO.pdf>
- Choi, H.-T., Lee, H.-J., Kang, H., Yu, S., & Park, H.-H. (2021). SSD-EMB: An Improved SSD Using Enhanced Feature Map Block for Object Detection. *Sensors*, 21(8), 2842. <https://doi.org/10.3390/s21082842>

- Cornell CALS. (2021, December 6). *Weed Management Methods*. <https://cals.cornell.edu/field-crops/corn/weed-control-corn/weed-management-methods>
- Dang, F., Chen, D., Lu, Y., & Li, Z. (2023). YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. *Computers and Electronics in Agriculture*, 205. <https://doi.org/10.1016/j.compag.2023.107655>
- Data Science. (2023, December 14). *Kaggle: todo lo que hay que saber sobre esta plataforma*. <https://datascientest.com/es/kaggle-todo-lo-que-hay-que-saber-sobre-esta-plataforma>
- DJI. (2021). *Mavic 2 - Product Information - DJI*. DJI. <https://www.dji.com/mavic-2/info>
- EDIFARM & CÍA. (2018). *MANUAL DE CULTIVO PASO A PASO*. https://gestion.edifarm.com.ec/edifarm_quickagro/pdfs/manual_cultivos/PAPA.pdf
- edX. (2022, July 8). *Deep Learning with Tensorflow*. EdX. <https://learning.edx.org/course/course-v1:IBM+DL0120EN+2T2021/home>
- Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., & Vasilakoglou, I. (2020). Towards weeds identification assistance through transfer learning. *Computers and Electronics in Agriculture*, 171, 105306. <https://doi.org/10.1016/j.compag.2020.105306>
- Espinosa Zúñiga, J. J. (2020). Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. *Ingeniería Investigación y Tecnología*, 21(1), 1-13. <https://doi.org/10.22201/ifi.25940732e.2020.21n1.008>
- Fatima, H. S., ul Hassan, I., Hasan, S., Khurram, M., Stricker, D., & Afzal, M. Z. (2023). Formation of a Lightweight, Deep Learning-Based Weed Detection System for a Commercial Autonomous Laser Weeding Robot. *Applied Sciences*, 13(6), 3997. <https://doi.org/10.3390/app13063997>
- García-Navarrete, O. L., Correa-Guimaraes, A., & Navas-Gracia, L. M. (2024). *Application of Convolutional Neural Networks in Weed Detection and Identification: A Systematic Review*. <https://doi.org/10.20944/preprints202401.1148.v1>
- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*.
- Gonzales-Martínez, R., Machacuay, J., Rotta, P., & Chinguel, C. (2022). Hyperparameters Tuning of Faster R-CNN Deep Learning Transfer for Persistent Object Detection in Radar Images. *IEEE Latin America Transactions*, 20(4), 677-685. <https://doi.org/10.1109/TLA.2022.9675474>
- Google. (2024). *Google Colaboratory*. <https://colab.google/>
- Google Cloud. (2022, December 4). *Intro to TensorFlow en Español*. Coursera. <https://es.coursera.org/learn/intro-tensorflow-es?>

- Government of New Brunswick. (2019). *Weed Control in Potato*. <https://www2.gnb.ca/content/dam/gnb/Departments/10/pdf/Agriculture/WeedControlPotato.pdf>
- Gradio. (2024). *Gradio Banner Build & share delightful machine learning apps*. <https://www.gradio.app/>
- Grimstad, B., & Andersson, H. (2019). *ReLU Networks as Surrogate Models in Mixed-Integer Linear Programs*. <https://doi.org/10.1016/j.compchemeng.2019.106580>
- Harvard University. (2015, September 5). *Unmanned Aircraft Systems / Drones*. Risk Management and Audit Services. <https://rmas.fad.harvard.edu/unmanned-aircraft-systems-drones>
- Hasan, A. S. M. M., Sohel, F., Diepeveen, D., Laga, H., & Jones, M. G. K. (2021). A survey of deep learning techniques for weed detection from images. *Computers and Electronics in Agriculture*, 184, 106067. <https://doi.org/10.1016/j.compag.2021.106067>
- HOTZ, N. (2023). What is CRISP DM? *Data Science Process Alliance*. <https://www.datascience-pm.com/crisp-dm-2/>
- Huo, B., Li, C., Zhang, J., Xue, Y., & Lin, Z. (2023). SAFF-SSD: Self-Attention Combined Feature Fusion-Based SSD for Small Object Detection in Remote Sensing. *Remote Sensing*, 15(12), 3027. <https://doi.org/10.3390/rs15123027>
- IBM. (2021). *CRISP-DM overview*. IBM. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>
- IBM. (2023a, February). *¿Qué es el aprendizaje supervisado?*. IBM. <https://www.ibm.com/es-es/topics/supervised-learning>
- IBM. (2023b, March 26). *Deep Learning Fundamentals with Keras*. EdX. <https://www.edx.org/course/deep-learning-fundamentals-with-keras>
- IBM. (2023, January 31). *¿Qué es el etiquetado de datos?* <https://www.ibm.com/es-es/topics/data-labeling>
- IBM. (2024). *¿Qué es el descenso del gradiente?* <https://www.ibm.com/mx-es/topics/gradient-descent>
- IBM Corporation. (2021, August 17). *Conceptos básicos de ayuda de CRISP-DM*. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>
- INIAP. (2014). *Manejo Agronómico*. <http://tecnologia.iniap.gob.ec/images/rubros/contenido/papa/malezas1.pdf>
- INIAP. (2018). *Control de malezas*. <https://eva.iniap.gob.ec/web2/papa/malezas-papa/>

- ISO. (2014). *International Standar ISO/IEC 25051*. <https://cdn.standards.iteh.ai/samples/61579/21f65a5556a0f430a9c6bd07729ae38ae/ISO-IEC-25051-2014.pdf>
- ISO 25000. (2022). *ISO 25010 calidad de software y datos*. <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- Jiang, Y., Li, X., Luo, H., Yin, S., & Kaynak, O. (2022). Quo vadis artificial intelligence? *Discover Artificial Intelligence*, 2(1), 4. <https://doi.org/10.1007/s44163-022-00022-8>
- Jiménez, S., & Merino, A. (2023). ARTICLE HISTORY CRISP-DM based Machine Learning Models for Analyzing the Depression Level in Students of the National Polytechnic School. *ARTICLE HISTORY LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC)*, IX(1), 2023. <https://doi.org/10.5281/zenodo.7503909>
- KeepCoding. (2021, June 7). *Regularización L2 en Red Neuronal Convolutacional*. KeepCoding. <https://keepcoding.io/blog/regularizacion-l2-red-neuronal-convolutacional/>
- keras. (2020, April 15). *Transfer learning and fine-tuning*. Keras. https://keras.io/guides/transfer_learning/#introduction
- Keras. (2021). *VGG16 function*. <https://keras.io/api/applications/vgg/#vgg16-function>
- keras. (2021, October 2). *MaxPooling2D layer*. Keras. https://keras.io/api/layers/pooling_layers/max_pooling2d/
- Khan, M. A., Akram, T., Sharif, M., Javed, K., Raza, M., & Saba, T. (2020). An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection. *Multimedia Tools and Applications*, 79(25-26), 18627-18656. <https://doi.org/10.1007/s11042-020-08726-8>
- Krishnaswamy, V., Singh, N., Sharma, M., Verma, N., & Verma, A. (2023). Application of CRISP-DM methodology for managing human-wildlife conflicts: an empirical case study in India. *Journal of Environmental Planning and Management*, 66(11), 2247-2273. <https://doi.org/10.1080/09640568.2022.2070460>
- Kukačka, J., Golkov, V., & Cremers, D. (2017). *Regularization for Deep Learning: A Taxonomy*.
- Li, F.-F., Johnson, J., & Yeung Serena. (2021). Convolutional Neural Networks for Visual Recognition. *Stanford University*. <https://cs231n.github.io/convolutional-networks/#layers>
- Li, J., Zhang, W., Zhou, H., Yu, C., & Li, Q. (2024). Weed detection in soybean fields using improved YOLOv7 and evaluating herbicide reduction efficacy. *Frontiers in Plant Science*, 14. <https://doi.org/10.3389/fpls.2023.1284338>
- Lingenfelter, D. (2022, June 1). *Pokeweed Control in Corn and Soybean*. <https://extension.psu.edu/pokeweed-control-in-corn-and-soybean>

- Liu, B., & Bruch, R. (2020). Weed Detection for Selective Spraying: a Review. *Current Robotics Reports*, 1(1), 19-26. <https://doi.org/10.1007/s43154-020-00001-w>
- Liu, S., Jin, Y., Ruan, Z., Ma, Z., Gao, R., & Su, Z. (2022). Real-Time Detection of Seedling Maize Weeds in Sustainable Agriculture. *Sustainability*, 14(22), 15088. <https://doi.org/10.3390/su142215088>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS. https://doi.org/10.1007/978-3-319-46448-0_2
- Lutkevich, B. (2021, December). *drone (UAV)*. TechTarget . <https://www.techtarget.com/iotagenda/definition/drone>
- Mahmudul Hasan, A. S. M., Sohel, F., Diepeveen, D., Laga, H., & Jones, M. G. K. (2022). Weed recognition using deep learning techniques on class-imbalanced imagery. *Crop and Pasture Science*. <https://doi.org/10.1071/CP21626>
- Majib, M. S., Rahman, M. M., Shahriar Sazzad, T. M., Khan, N. I., & Dey, S. K. (2021). VGG-SCNet: A VGG Net based Deep Learning framework for Brain Tumor Detection on MRI Images. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2021.3105874>
- Manasa, K., & Murthy, D. G. V. (2021). Skin Cancer Detection Using VGG-16. *European Journal of Molecular & Clinical Medicine*, 8(1), 1419-1426.
- Matt Russell. (2018). Deep Learning for Computer Vision. In K. R & B. P (Eds.), *Deep Learning for Computer Vision* (pp. 81-107). O'Reilly Media. <https://www.oreilly.com/library/view/deep-learning-for/9781788295628/1ab53400-64ab-43d7-9088-ad64943351f9.xhtml>
- Microsoft. (2023, November 15). *The new Microsoft Planner: A unified experience bringing together to-dos, tasks, plans and projects*. <https://techcommunity.microsoft.com/t5/planner-blog/the-new-microsoft-planner-a-unified-experience-bringing-together/ba-p/3977998>
- Moazzam, S. I., Khan, U. S., Qureshi, W. S., Tiwana, M. I., Rashid, N., Alasmay, W. S., Iqbal, J., & Hamza, A. (2021). A Patch-Image Based Classification Approach for Detection of Weeds in Sugar Beet Crop. *IEEE Access*, 9, 121698-121715. <https://doi.org/10.1109/ACCESS.2021.3109015>
- Moncayo, S., & Karla, F. (2021). *Visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales* [Universidad Técnica del Norte]. <http://repositorio.utn.edu.ec/handle/123456789/11567>
- Moscocio, E. (2020). *Clasificador de Señales de Tráfico Basado en Redes Neuronales Convolucionales para Robots Móviles*. <https://1library.co/article/visual-geometry-group-vgg-arquitecturas-famosas-redes-convolucionales.yn002xkq>

- New England Institute of Technology. (2022, February 22). *The 20 Best UAV Schools/Colleges with Drone Programs in 2023*. <https://www.neit.edu/blog/uav-schools#:~:text=Top%20Unmanned%20Aircraft%20Systems%2FDrone%20Training%20Colleges%3A%20Accredited%201,Michigan%20College%2C%20Traverse%20City%2C%20Michigan%20...%20M%C3%A1s%20elementos>
- Niu, L., Liu, J., Wang, X., Liu, W., Yang, D., & Yuan, Z. (2023). *Improving Post-Training Quantization on Object Detection with Task Loss-Guided L_p Metric*.
- Ofori, M., & El-Gayar, O. (2021). *An Approach for Weed Detection Using CNNs And Transfer Learning*. <https://doi.org/10.24251/HICSS.2021.109>
- Oktavius, A. K., Manalu, S. R., Sasmoko, Indrianti, Y., & Moniaga, J. V. (2022). Artificial Intelligence in Entrepreneurial Mindfulness Using CRISP-DM Method. *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, 1-7. <https://doi.org/10.1109/ICITDA55840.2022.9971384>
- Oliveros, J. (2022). *Fotogrametría aérea basada en rpas para la caracterización de fanerógamas marinas en el fondo de ABADES, TENERIFE* [COOPERACIÓN AL DESARROLLO, Universidad Politécnica de Madrid]. https://oa.upm.es/70549/1/TFM_Juan_Alfonso_Oliveros_Lopez_2.pdf
- Ormeño, N. (2019, May 15). *ISO 25010 y el desarrollo de software*. <https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341>
- Pravitasari, A. A., Iriawan, N., Almuhayar, M., Azmi, T., Irhamah, Fithriasari, K., Purnami, S. W., & Ferriastuti, W. (2020). UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(3), 1310-1318. <https://doi.org/10.12928/TELKOMNIKA.v18i3.14753>
- PyTorch. (2021). *Tensors and dynamic neural networks in Python with strong GPU acceleration*. <https://pytorch.org/>
- Robayo, A., & García, I. (2022). *Construcción de un sistema de visión por computadora para la detección de maleza y líneas de cultivo utilizando imágenes provenientes de un Dron* [Universidad Técnica del Norte]. <http://repositorio.utn.edu.ec/handle/123456789/11950>
- Roboflow. (2020). *Build Vision Models with Roboflow*. <https://docs.roboflow.com/>
- Roboflow. (2024). *Computer vision tools for developers and enterprises*. <https://roboflow.com/>
- Rodríguez, E. (2020, September 13). *Malezas - Concepto, origen, evolución y efectos*. *Lancet* (London, England). <https://www.quifuca.com/ve/2020/09/13/malezas-concepto-origen-evolucion-y-efectos/>
- Sahoo, P. K., Panda, M. K., Panigrahi, U., Panda, G., Jain, P., Islam, Md. S., & Islam, M. T. (2024). An Improved VGG-19 Network Induced Enhanced Feature Pooling for

- Precise Moving Object Detection in Complex Video Scenes. *IEEE Access*, 12, 45847-45864. <https://doi.org/10.1109/ACCESS.2024.3381612>
- Saleh, A., Olsen, A., Wood, J., Philippa, B., & Azghadi, M. R. (2024). *Semi-Supervised Weed Detection for Rapid Deployment and Enhanced Efficiency*.
- Sánchez, C., & Calderón, A. (2020). *Procesamiento digital de imágenes obtenidas con dron mediante el uso de software libre* [Fotogrametría, Universidad Distrital Francisco José de Caldas]. <https://repository.udistrital.edu.co/bitstream/handle/11349/26169/SanchezArizaCarlosIvan2020.pdf?isAllowed=y&sequence=1>
- Schwartz-Lazaro, L. M., Gage, K. L., & Chauhan, B. S. (2021). Editorial: Weed Biology and Ecology in Agroecosystems. In *Frontiers in Agronomy* (Vol. 3). Frontiers Media S.A. <https://doi.org/10.3389/fagro.2021.730074>
- Seiche, A. T., Wittstruck, L., & Jarmer, T. (2024). Weed Detection from Unmanned Aerial Vehicle Imagery Using Deep Learning—A Comparison between High-End and Low-Cost Multispectral Sensors. *Sensors*, 24(5), 1544. <https://doi.org/10.3390/s24051544>
- Sharma, T., Debaque, B., Duclos, N., Chehri, A., Kinder, B., & Fortier, P. (2022). Deep Learning-Based Object Detection and Scene Perception under Bad Weather Conditions. *Electronics*, 11(4), 563. <https://doi.org/10.3390/electronics11040563>
- Shewchuk, J. (2020). Introduction to Machine Learning. *HW4*. <https://people.eecs.berkeley.edu/~jrs/189s20/hw/hw4.pdf>
- Simon Buehring, S. (2021, July 21). *The 4 Kanban principles: A visual guide*. Knowledge Train. <https://www.knowledgetrain.co.uk/agile/agile-project-management/agile-project-management-course/kanban-principles>
- Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. <https://arxiv.org/abs/1409.1556v6>
- Su, W.-H. (2020). Crop plant signaling for real-time plant identification in smart farm: A systematic review and new concept in artificial intelligence for automated weed control. *Artificial Intelligence in Agriculture*, 4, 262-271. <https://doi.org/10.1016/J.AIIA.2020.11.001>
- Subeesh, A., Bhole, S., Singh, K., Chandel, N. S., Rajwade, Y. A., Rao, K. V. R., Kumar, S. P., & Jat, D. (2022). Deep convolutional neural network models for weed detection in polyhouse grown bell peppers. *Artificial Intelligence in Agriculture*, 6, 47-54. <https://doi.org/10.1016/J.AIIA.2022.01.002>
- Teamhood. (2022). 4 Core Kanban Principles and 6 Practices . *Teamhood*. <https://teamhood.com/kanban-resources/kanban-principles/>
- TensorFlow. (2022). keras. *TensorFlow*.

- TensorFlow. (2023, March 14). *Introduction to TensorFlow*. TensorFlow. <https://www.tensorflow.org/learn?hl=es-419>
- Torres, J. (2020). *PYTHON DEEP LEARNING Introducción práctica con Keras y TensorFlow 2*.
- Unipython. (2021, March 5). *INTRODUCCIÓN Y COMO INSTALAR KERAS EN PYTHON ANACONDA*. Unipython.
- Villalba, M. (2020, October 26). *Arquitectura VGG16 y VGG19 en Deep Learning*. <https://keepcoding.io/blog/arquitectura-vgg16-vgg19-deep-learning/>
- VISO AI. (2021). VGG Very Deep Convolutional Networks (VGGNet). *VISO AI*. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>
- Wang, X., Kong, L., Zhang, Z., Wang, H., & Lu, X. (2023). Keypoint regression strategy and angle loss based YOLO for object detection. *Scientific Reports*, *13*(1), 20117. <https://doi.org/10.1038/s41598-023-47398-w>
- Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing*, *2019*(1), 40. <https://doi.org/10.1186/s13640-019-0417-8>
- Yam, E. (2022). *Low-Rank and Temporal Smoothness Regularization on Value-Based Deep Reinforcement Learning*. <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-218.html>