



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE

**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR,
MODALIDAD TRABAJO DE GRADO**

TEMA:

**“ENTRENAMIENTO DE LA RED NEURONAL CONVOLUCIONAL U-
NET PARA LA CUANTIFICACIÓN AUTOMÁTICA DE MALEZAS EN
CAMPOS DE CULTIVOS DE MAÍZ Y/O PAPA, UTILIZANDO
IMÁGENES ADQUIRIDAS POR UN DRON”**

Trabajo de titulación previo a la obtención del título de INGENIERO EN SOFTWARE

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética).

AUTOR:

Jorge Ramiro Pazos Morillo

DIRECTOR:

PhD. Iván Danilo García Santillán

Ibarra, enero 2025



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100317477-6		
APELLIDOS Y NOMBRES:	Pazos Morillo Jorge Ramiro		
DIRECCIÓN:	13 de Abril y Guayas 2-80		
EMAIL:	jrpazosm@utn.edu.ec		
TELÉFONO FIJO:	2546904	TELÉFONO MÓVIL:	0939884327

DATOS DE LA OBRA	
TÍTULO:	Entrenamiento de la red neuronal convolucional U-Net para la cuantificación automática de malezas en campos de cultivos de maíz y/o papa, utilizando imágenes adquiridas por un dron
AUTOR (ES):	Pazos Morillo Jorge Ramiro
FECHA DE APROBACIÓN: DD/MM/AAAA	14/01/2025
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Software
DIRECTOR:	PhD. Iván Danilo García Santillán
ASESOR:	PhD. Marco Remigio Pusda Chulde

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 14 días del mes de enero de 2025

EL AUTOR:

(Firma).....

Nombre: Pazos Morillo Jorge Ramiro

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 14 de enero de 2025,

PhD. Iván Danilo García Santillán

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f)

PhD. Iván Danilo García Santillán

C.C.: 1002292603

Dedicatoria

Este trabajo de grado está dedicado a mis queridos padres y hermanos, su inquebrantable fe en mí ha sido mi mayor impulso a lo largo de este arduo camino académico. Vuestro constante apoyo, amor incondicional y sacrificios han sido la base de mi éxito. Sin ustedes, este logro no sería posible.

A mis amigos y compañeros de universidad, que con su amistad, complicidad y aliento han iluminado cada paso de este viaje. Vuestras palabras de ánimo, las largas noches de estudio compartidas en Discord y las risas que hemos compartido han hecho que este camino sea más llevadero y memorable.

A mi sobrinito Esteban, quien partió demasiado pronto, tu presencia fugaz pero impactante en nuestras vidas sigue siendo una fuente de inspiración. Aunque ya no estés físicamente entre nosotros, tu recuerdo y tu espíritu vivirán eternamente en nuestros corazones.

Este logro no solo es mío, sino también de aquellos que me han acompañado y guiado a lo largo de esta travesía.

Agradecimientos

Quisiera expresar mi sincero agradecimiento a todas las personas que contribuyeron de alguna manera a la realización de este trabajo académico.

En primer lugar, quiero agradecer a mis padres y hermanos por su amor incondicional y su apoyo inquebrantable que me ha impulsado a superar desafíos y alcanzar mis metas. Su constante aliento y ejemplo de trabajo duro han sido una inspiración para mí en cada paso de este viaje.

A mi sobrinito Esteban, aunque su tiempo entre nosotros fue breve, tu presencia dejó una huella imborrable en nuestras vidas. Tu recuerdo continuará inspirándome siempre.

A mis amigos y compañeros de universidad, agradezco su compañerismo, risas compartidas e inolvidables recuerdos que siempre estarán presentes en mí.

Quiero dedicar un agradecimiento especial a todas esas chicas que estuvieron a mi lado y me brindaron su amor durante este trayecto. En particular, a una persona muy especial, cuya presencia iluminó mis días y me inspiró a ser una mejor versión de mí mismo. Su apoyo ha sido esencial en este viaje.

A mi querida mejor amiga y a mi mejor amigo, su amistad ha sido un pilar fundamental en mi camino académico. Agradezco su constante apoyo, comprensión y aliento. Ambos han sido mi fortaleza en tiempos difíciles y mi motivación para seguir adelante. Este logro es también un reflejo de nuestra amistad, que atesoraré por siempre.

Finalmente, quiero expresar mi gratitud a mis profesores y asesores académicos por su orientación, conocimientos y valiosas sugerencias que han enriquecido este trabajo. Su dedicación y pasión por la enseñanza han sido una inspiración constante.

Tabla de contenido

Certificación del Director del Trabajo de Grado	IV
Dedicatoria	V
Agradecimientos	VI
Tabla de contenido	VII
Índice de figuras	IX
Índice de tablas	XI
Resumen.....	XIII
Abstract	XIV
Introducción	XV
Planteamiento del problema	XV
Objetivos	XVI
Objetivo general	XVI
Objetivo específico.....	XVI
Alcance y metodología	XVI
Alcance	XVI
Metodología	XX
Justificación.....	XXII
Riesgos.....	XXIV
CAPÍTULO 1	1
1.1. Contexto General	1
1.1.1. Agricultura y detección de malezas.....	1
1.1.2. Tecnologías para la cuantificación automática de malezas.....	2
1.2. Fundamentos de las redes neuronales convolucionales (CNN)	4
1.2.1. Introducción a las redes neuronales convolucionales.....	4
1.2.2. Redes neuronales convolucionales en la detección de malezas.....	5
1.3. La red neuronal convolucional U-Net	6
1.3.1. Introducción a U-Net	6
1.3.2. Arquitectura y funcionamiento de U-Net	8
1.3.3. Entrenamiento y ajuste de U-Net.....	10
1.4. Metodología KDD	11
1.5. Evaluación de la arquitectura de U-Net	13
1.5.1. Métricas de evaluación para la segmentación semántica	13
TRABAJOS RELACIONADOS	15

CAPÍTULO 2	18
2.1. Descripción del proyecto	18
2.1.1. Planificación de actividades	18
2.1.2. Entregables	19
2.2. Recolección de datos	20
2.3. Selección de imágenes	31
2.4. Preprocesamiento y transformación de datos	34
2.5. Anotación manual	36
2.6. Aumentación de datos	40
2.7. Minería de datos	41
2.7.1. Implementación del algoritmo U-Net	42
2.7.2. Entrenamiento del modelo	45
2.7.3. Resultados del entrenamiento	46
2.7.4. Variaciones del modelo	48
2.8. Interpretación de datos	54
2.9. Despliegue del modelo	58
2.9.1. Backend	59
2.9.2. Frontend	61
2.9.3. Dockerización	64
 CAPÍTULO 3	 65
3.1. Comparativa de resultados de entrenamientos del modelo U-Net	65
3.1.1. Comparación en conjunto	82
3.1.2. Análisis de resultados y elección del mejor modelo	87
3.2. Validación según la Norma ISO 25023	87
3.3. Comparación de tiempos entre el método manual y el propuesto	91
 Discusión	 95
Conclusiones	106
Recomendaciones	108
Referencias	109
Anexos	113

Índice de figuras

<i>Figura 1</i> Árbol de problemas	XIV
<i>Figura 2</i> Arquitectura de la red neuronal convolucional U-NET.	XVII
<i>Figura 3</i> Arquitectura de la aplicación móvil	XVIII
<i>Figura 4</i> Metodología	XIX
<i>Figura 5</i> Metodología KDD (Knowledge Discovery in Databases)	12
<i>Figura 6</i> Actividades para la recolección de datos.....	20
<i>Figura 7</i> Terreno n°8 adecuado.....	22
<i>Figura 8</i> Siembra de malezas en el terreno n°8	23
<i>Figura 9</i> Aterrizaje del dron DJI Mavic 2 Pro	24
<i>Figura 10</i> Imagen capturada en el terreno 1 en la primera visita el 20/05/2023.....	25
<i>Figura 10</i> Imagen capturada en el terreno 1 en la segunda visita el 28/05/2023	26
<i>Figura 12</i> Imagen capturada en el terreno 1 en la tercera visita el 04/06/2023.....	26
<i>Figura 13</i> Imagen capturada en el terreno 2 el 04/06/2023	27
<i>Figura 14</i> Imagen capturada en el terreno 3 el 04/06/2023	28
<i>Figura 15</i> Imagen capturada en el terreno 4 el 16/12/2023	28
<i>Figura 16</i> Imagen capturada en el terreno 5 el 16/12/2023	29
<i>Figura 17</i> Imagen capturada en el terreno 6 el 16/12/2023	29
<i>Figura 18</i> Imagen capturada en el terreno 7 el 16/12/2023	30
<i>Figura 19</i> Imagen capturada en el terreno 8 en la primera visita el 11/05/2024.....	30
<i>Figura 20</i> Imagen capturada en el terreno 8 en la segunda visita el 17/05/2024	31
<i>Figura 21</i> Actividades para la etapa de selección de imágenes	32
<i>Figura 22</i> Imágenes que fueron descartadas en la fase de selección	33
<i>Figura 23</i> Extracción de subimágenes de 250 x 250 píxeles.....	35
<i>Figura 24</i> Actividades para la anotación manual en Roboflow.....	36
<i>Figura 25</i> Imagen anotada en Roboflow.....	38
<i>Figura 26</i> Imagen con su máscara de segmentación semántica generada	38
<i>Figura 27</i> Actividades para la minería de datos.....	42
<i>Figura 28</i> Arquitectura U-Net.....	43
<i>Figura 29</i> Visualización de datos de entrenamiento	46
<i>Figura 30</i> Arquitectura de U-Net Residual.....	49
<i>Figura 31</i> Arquitectura de U-Net Doble	51
<i>Figura 32</i> Arquitectura de Modified U-Net	52
<i>Figura 33</i> Arquitectura de U-NetAtt	54
<i>Figura 34</i> Visualización de segmentación en imagen de 128 x 128 píxeles	55
<i>Figura 35</i> Visualización de segmentación de imagen completa 5472 x 3648 píxeles	57

<i>Figura 36 Arquitectura del sistema</i>	59
<i>Figura 37 Flujo de trabajo de predicción del backend</i>	60
<i>Figura 38 Pantalla de bienvenida</i>	62
<i>Figura 39 Pantalla de inferencias</i>	63
<i>Figura 40 Flujo de trabajo de la pantalla de inferencia</i>	64
<i>Figura 41 Gráfico de precisión U-Net</i>	66
<i>Figura 42 Gráfico de pérdida U-Net</i>	67
<i>Figura 43 Gráfico de precisión U-Net 100 épocas</i>	67
<i>Figura 44 Gráfico de pérdida U-Net 100 épocas</i>	68
<i>Figura 45 Resultado de predicción y máscara real con U-Net</i>	69
<i>Figura 46 Gráfico de pérdida de U-Net Residual</i>	70
<i>Figura 47 Gráfico de precisión de U-Net Residual</i>	70
<i>Figura 48 Gráfico de pérdida de U-Net Residual 100 épocas</i>	71
<i>Figura 49 Gráfico de precisión de U-Net Residual 100 épocas</i>	71
<i>Figura 50 Resultado de predicción y máscara real con U-Net Residual</i>	73
<i>Figura 51 Gráfico de pérdida de U-Net Doble</i>	74
<i>Figura 52 Gráfico de precisión de U-Net Doble</i>	74
<i>Figura 53 Gráfico de pérdida de U-Net Doble 100 épocas</i>	75
<i>Figura 54 Gráfico de precisión de U-Net Doble 100 épocas</i>	75
<i>Figura 55 Resultado de predicción y máscara real con U-Net Doble</i>	77
<i>Figura 56 Gráfico de pérdida de MU-Net</i>	77
<i>Figura 57 Gráfico de precisión de MU-Net</i>	78
<i>Figura 58 Resultado de predicción y máscara real con MU-Net</i>	79
<i>Figura 59 Gráfico de pérdida de U-NetAtt</i>	80
<i>Figura 60 Gráfico de precisión de U-NetAtt</i>	80
<i>Figura 61 Resultado de predicción y máscara real con U-NetAtt</i>	82
<i>Figura 62 Gráficas de comparación de precisión y pérdida</i>	83
<i>Figura 63 Validación sobre la subcaracterística Comportamiento del tiempo de la ISO 25023</i> . 88	
<i>Figura 64 Distribución de tiempos de respuesta</i>	89
<i>Figura 65 Tiempos de respuesta por solicitud</i>	90
<i>Figura 66 Promedio y desviación estándar de los tiempos de respuesta</i>	91

Índice de tablas

Tabla 1 <i>Riesgos del proyecto</i>	XXII
Tabla 2 <i>Matriz de riesgos</i>	XXIII
Tabla 3 <i>Trabajos relacionados</i>	15
Tabla 4 <i>Planificación</i>	18
Tabla 5 <i>Entregables</i>	19
Tabla 6 <i>Terrenos visitados</i>	21
Tabla 7 <i>Imágenes obtenidas por cada terreno</i>	24
Tabla 8 <i>Imágenes por cada clase</i>	33
Tabla 9 <i>Imágenes seleccionadas para balancear las clases</i>	34
Tabla 10 <i>Resumen del preprocesamiento de imágenes</i>	36
Tabla 11 <i>Clases definidas por cada tipo de maleza y cultivo</i>	37
Tabla 12 <i>Número de imágenes por clase para el conjunto de entrenamiento y validación</i>	39
Tabla 13 <i>Número de individuos por cada clase en comparación con el dataset original</i>	39
Tabla 14 <i>Técnicas de Data Augmentation y número de imágenes obtenidas</i>	40
Tabla 15 <i>Número de individuos por cada clase en el dataset balanceado</i>	41
Tabla 16 <i>Resumen de la arquitectura del modelo U-Net</i>	43
Tabla 17 <i>Especificaciones de las GPUs en Google Colab</i>	45
Tabla 18 <i>Parámetros del modelo entrenado</i>	47
Tabla 19 <i>Parámetros de entrenamiento de U-Net Residual</i>	50
Tabla 20 <i>Parámetros de entrenamiento de U-Net Doble</i>	51
Tabla 21 <i>Parámetros de entrenamiento de U-Net Doble</i>	53
Tabla 22 <i>Parámetros de entrenamiento de U-NetAtt</i>	54
Tabla 23 <i>Porcentaje de píxeles por clase en la máscara real y en la predicción</i>	56
Tabla 24 <i>Porcentaje de píxeles por clase en imagen completa 5472 x 3648 píxeles</i>	57
Tabla 25 <i>Dependencias instaladas</i>	59
Tabla 26 <i>Paquetes NuGet instalados</i>	61
Tabla 27 <i>Métricas de evaluación del modelo U-Net</i>	65
Tabla 28 <i>Resultados mean dice coefficient, dice loss y mean IoU de U-Net</i>	68
Tabla 29 <i>Resultados mean IoU para cada clase U-Net</i>	69
Tabla 30 <i>Tiempos de inferencia de U-Net</i>	69
Tabla 31 <i>Resultados mean dice coefficient, dice loss y mean IoU de U-Net Residual (dataset desbalanceado)</i>	72
Tabla 32 <i>Resultados mean dice coefficient, dice loss y mean IoU de U-Net Residual (dataset balanceado)</i>	72

Tabla 33 Resultados mean IoU para cada clase U-Net Residual (dataset desbalanceado)	72
Tabla 34 Resultados mean IoU para cada clase U-Net Residual (dataset balanceado)	73
Tabla 35 Tiempos de inferencia de U-Net Residual	73
Tabla 36 Resultados mean dice coefficient, dice loss y mean IoU de U-Net Doble	76
Tabla 37 Resultados mean IoU para cada clase U-Net Doble.....	76
Tabla 38 Tiempos de inferencia de U-Net Doble	76
Tabla 39 Resultados mean dice coefficient, dice loss y mean IoU de MU-Net	78
Tabla 40 Resultados mean IoU para cada clase MU-Net.....	79
Tabla 41 Tiempos de inferencia de MU-Net	79
Tabla 42 Resultados mean dice coefficient, dice loss y mean IoU de U-NetAtt.....	81
Tabla 43 Resultados mean IoU para cada clase U-NetAtt.....	81
Tabla 44 Tiempos de inferencia de U-NetAtt.....	81
Tabla 45 Comparativa de resultados en el conjunto de entrenamiento y validación	82
Tabla 46 Comparativa de resultados dice loss, mean dice coefficient y mean IoU (entrenamiento).....	83
Tabla 47 Comparativa de resultados dice loss, mean dice coefficient y mean IoU (conjunto validación).....	84
Tabla 48 Resultados mean IoU para cada clase para todos los modelos.....	86
Tabla 49 Resultados de respuestas HTTP.....	88
Tabla 50 Tiempo empleado para identificar y cuantificar malezas en los 3 terrenos seleccionados de forma tradicional	92
Tabla 51 Tiempo empleado para identificar y cuantificar malezas en los 3 terrenos seleccionados usando el aplicativo móvil	92
Tabla 52 Prueba t Student a los conjuntos de tiempo TT y TA.....	93
Tabla 53 Resultados de trabajos relacionados enfocados en 2 clases.....	95
Tabla 54 Resultados de trabajos relacionados enfocados en 3 clases.....	96
Tabla 55 Resultados de trabajos relacionados enfocados en las mismas clases utilizadas	97
Tabla 56 Comparativa con trabajos relacionados enfocados en 2 clases (cultivo, maleza)	98
Tabla 57 Comparativa con trabajos relacionados enfocados en 3 clases (Background, cultivo, maleza)	101
Tabla 58 Comparativa con trabajos relacionados enfocados en las mismas clases utilizadas	103

Resumen

En el presente trabajo se ha implementado un modelo basado en redes neuronales convolucionales, específicamente la arquitectura U-Net y sus variaciones, para la segmentación automática de malezas en cultivos agrícolas, utilizando imágenes adquiridas por drones. La investigación parte de la problemática asociada a la cuantificación manual de malezas, un proceso que suele ser subjetivo, laborioso y dependiente de factores como la experiencia del agricultor y las condiciones del terreno. Este proyecto propone una solución automatizada que combina avances tecnológicos en visión por computadora y aprendizaje profundo, con el objetivo de optimizar la precisión y eficiencia en el manejo de cultivos.

El modelo se entrenó y evaluó utilizando un conjunto de datos balanceado, que incluye las clases: background (fondo), lengua de vaca (Cow-tongue), diente de león (Dandelion), kikuyo (Kikuyo), otras malezas (Other), y papa (Potato). Se desarrollaron y evaluaron cinco variaciones del modelo U-Net: U-Net original, U-Net Residual, U-Net Doble, MU-Net y U-Net con módulo de atención y bloques residuales. Las imágenes utilizadas fueron sometidas a técnicas de preprocesamiento y aumento de datos para balancear las clases y mejorar la representación de malezas subrepresentadas. Además, se diseñó un conjunto de métricas (Mean Dice Coefficient, Mean IoU y Dice Loss) para evaluar el rendimiento de cada modelo.

Los resultados obtenidos destacaron a la U-Net Residual como la mejor variación, con un desempeño óptimo en la segmentación de las clases objetivo. Adicionalmente, se validó la solución mediante una comparativa del tiempo de identificación y cuantificación de malezas, demostrando que el sistema desarrollado reduce significativamente el tiempo necesario en comparación con los métodos manuales tradicionales. Finalmente, se desplegó la arquitectura seleccionada en una API REST dockerizada para facilitar su integración en una aplicación móvil, promoviendo así su adopción práctica en el sector agrícola.

Este trabajo constituye un aporte significativo al campo de la agricultura de precisión, demostrando que las redes neuronales convolucionales, combinadas con drones y herramientas tecnológicas, pueden revolucionar el manejo de cultivos, promoviendo prácticas agrícolas más sostenibles y eficientes.

Abstract

This study implements a convolutional neural network-based model, specifically the U-Net architecture and its variations, for the automatic segmentation of weeds in agricultural crops using drone-acquired images. The research addresses the challenges of manual weed quantification, a process that is often subjective, labor-intensive, and dependent on factors such as farmer expertise and terrain conditions. This project proposes an automated solution that integrates advancements in computer vision and deep learning to optimize the accuracy and efficiency of crop management.

The model was trained and evaluated using a balanced dataset that includes the following classes: background, Cow-tongue, Dandelion, Kikuyo, Other weeds, and Potato. Five variations of the U-Net model were developed and evaluated: the original U-Net, Residual U-Net, Double U-Net, MU-Net, and U-Net with attention module and residual blocks. The images underwent preprocessing and data augmentation techniques to balance classes and enhance the representation of underrepresented weeds. A set of metrics (Mean Dice Coefficient, Mean IoU, and Dice Loss) was designed to assess the performance of each model.

The results highlighted the Residual U-Net as the best-performing variation, achieving optimal segmentation for the target classes. Additionally, the solution was validated through a comparison of the time required for weed identification and quantification, demonstrating that the developed system significantly reduces the time compared to traditional manual methods. The selected architecture was then deployed in a dockerized REST API to enable its integration into a mobile application, facilitating its practical adoption in the agricultural sector.

This work represents a significant contribution to the field of precision agriculture, demonstrating that convolutional neural networks, combined with drones and technological tools, can revolutionize crop management by promoting more sustainable and efficient agricultural practices.

Introducción

Planteamiento del problema

Según la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO, 2007), las plantas indeseables, también conocidas como malezas o malas hierbas, representan uno de los mayores desafíos para la producción agrícola a nivel mundial debido a que causan un daño económico y social al agricultor. El daño causado por estas plagas puede oscilar entre el 5% y el 10% de las cosechas en los países desarrollados, mientras que en los países en desarrollo y algunos de economía emergente, cuya producción agropecuaria es fundamental, las pérdidas pueden superar el 20% o incluso el 30% de la producción.

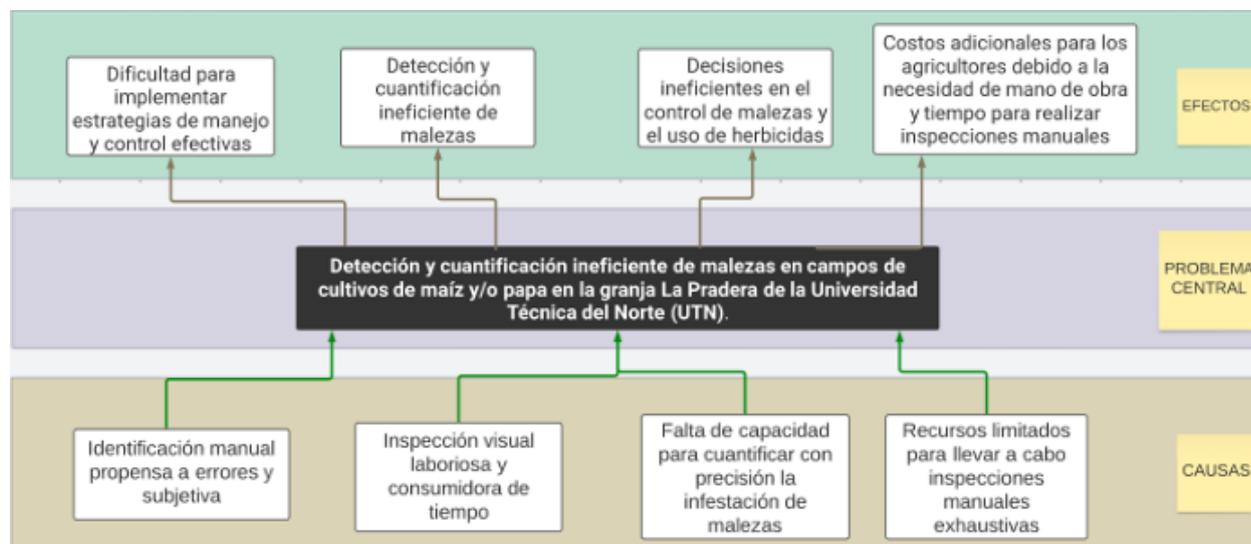
La identificación precisa de malezas en la agricultura desempeña un papel fundamental en el éxito de la agricultura y el manejo eficiente de los recursos, puesto que, en las plantaciones, la presencia de estas malas hierbas constituye una amenaza significativa para el rendimiento de los cultivos y la calidad de la cosecha. Según el estudio realizado por Craine & Dybzinski (2013) y el estudio de Pusedá-Chulde (2020), las malezas compiten por los recursos del suelo, como nutrientes y agua, afectando el crecimiento y desarrollo de las plantas deseadas. Además, algunas especies de malezas pueden liberar sustancias químicas tóxicas o actuar como hospederos de plagas y enfermedades, lo que agrava aún más los daños a los cultivos (Smith et al., 2020; Johnson et al., 2018).

A nivel local, se ha evidenciado una problemática en los campos de cultivos de maíz y/o papa en la granja La Pradera de la Universidad Técnica del Norte (UTN) al momento de realizar una cuantificación tradicional, lo que provoca que no se pueda implementar estrategias de manejo y control efectivas para la erradicación de estas malezas. Según estudios previos, se estima que la identificación manual de malezas puede llevar a errores significativos en la cuantificación, lo que resulta en decisiones poco eficaces en el control de malezas y el uso de herbicidas (Smith et al., 2019; Li et al., 2020). Asimismo, este enfoque puede resultar laborioso, ineficiente y propenso a errores, consumiendo una gran cantidad de tiempo y mano de obra, lo que implica costos adicionales para los agricultores, debido a que se requiere de una inspección minuciosa de los campos por parte de estos expertos agrícolas. Además, la inspección visual puede ser subjetiva y está sujeta a errores humanos, lo que resulta en una cuantificación inexacta de la infestación de malezas (Brown et al., 2019; Li et al., 2021).

La Figura 1 presenta el árbol de problemas del proyecto, el cual visualiza de manera gráfica las diversas problemáticas identificadas en el contexto de estudio.

Figura 1

Árbol de problemas



Nota. Elaboración propia

Objetivos

Objetivo general

- Entrenar una red neuronal convolucional U-Net para la cuantificación automática de malezas en campos de cultivos de maíz y/o papa, utilizando imágenes adquiridas por un dron.

Objetivo específico

- Realizar el marco teórico acerca de la red neuronal convolucional U-Net para la segmentación semántica de objetos.
- Adquirir y etiquetar imágenes propias usando el dron y una herramienta de anotación de imágenes.
- Desarrollar una aplicación móvil que utilice la arquitectura de la red neuronal convolucional U-Net para la identificación y cuantificación de malezas en los cultivos de maíz y/o papa.
- Validar los resultados como la precisión y la eficiencia del sistema propuesto utilizando métricas estadísticas y norma ISO 25023.

Alcance y metodología

Alcance

La finalidad del proyecto consiste en desarrollar un sistema de cuantificación automática de malezas en campos de cultivos de maíz y/o papa utilizando imágenes adquiridas por un dron y aplicando la red neuronal convolucional U-Net. El sistema incluirá la

adquisición y etiquetado de imágenes propias, el desarrollo de una aplicación móvil utilizando Xamarin y la evaluación de los resultados obtenidos. Se empleará la metodología KDD para el desarrollo del proyecto (Timarán Pereira et al., 2016).

La aplicación móvil se diseñará para utilizar la arquitectura de U-Net y permitirá a los usuarios usar imágenes de los campos de cultivo utilizando un dron. Estas imágenes serán offline previamente etiquetadas y luego procesadas por el modelo de U-Net entrenado para identificar y cuantificar automáticamente las malezas presentes en los cultivos de maíz y/o papa. El sistema proporcionará resultados de cuantificación precisos y eficientes, reduciendo el tiempo empleado en la clasificación de las malas hierbas.

Para lograrlo, se realizará un marco teórico sobre la red neuronal convolucional U-Net y su aplicación en la segmentación semántica de objetos. La arquitectura de U-Net, propuesta por Ronneberger et al. (2015), ha demostrado ser efectiva en tareas de segmentación y reconocimiento de objetos en imágenes médicas y de visión por computadora. Esto permitirá comprender los fundamentos teóricos y técnicos necesarios para implementar el modelo de U-Net en el desarrollo de la aplicación móvil (Krizhevsky et al., 2012). La Figura 2 muestra la arquitectura de la red neuronal convolucional U-Net (Ronneberger et al., 2015). Dicho marco teórico será en base a información obtenida de diferentes motores de búsqueda y bases de datos bibliográficas proporcionadas por la Universidad Técnica del Norte, además, se pretende responder las preguntas establecidas por el director del proyecto de tesis, el PhD. Iván Danilo García Santillán, las cuales se detallan a continuación:

- ¿Qué drones y cámaras están usando para la tarea de detección automática de malas hierbas?
- ¿Cuál es la altura y velocidad adecuada del dron para capturar imágenes y regiones de interés (ROIs) de alta resolución (px/cm)?
- ¿Qué espacio de color y formato de imágenes resulta adecuado?
- ¿En qué bandas del espectro electromagnético (visible, infrarrojo, ultravioleta) están trabajando?
- ¿Qué cultivos y tipos de malezas se están detectando y dónde?
- ¿Qué conjunto de datos se encuentran disponibles públicamente para el entrenamiento de un modelo de Deep Learning?
- ¿Cómo crear y anotar un conjunto de datos propio?
- ¿Qué arquitectura de CNN se usan para la detección automática de malas hierbas?
- ¿Qué hardware específico o servicios en la nube resulta adecuado para entrenar modelos de DL?

- ¿Qué métricas y gráficas se usan para validar el rendimiento de los modelos de DL?
- ¿Cómo afinar un modelo de DL para mejorar el rendimiento?

Se adquirirán imágenes propias utilizando un dron y se utilizará una herramienta de anotación de imágenes de software libre para etiquetarlas correctamente. Esto asegurará que las imágenes utilizadas para el entrenamiento y evaluación del modelo estén debidamente anotadas, lo que es fundamental para el aprendizaje automático y la precisión del sistema (García-García et al., 2018). Dichas imágenes se subirán a un repositorio el cuál contendrá un mínimo de 50 imágenes de los cultivos de maíz y/o papa en la granja La Pradera de la Universidad Técnica del Norte.

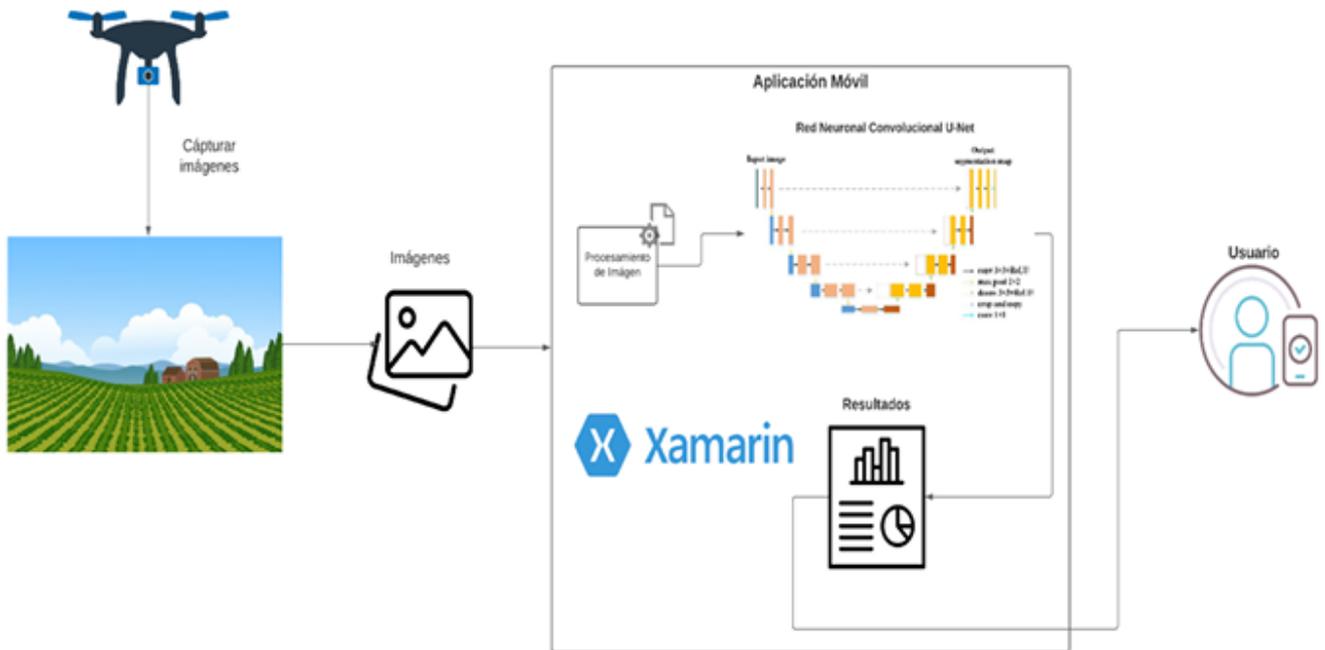
En el contexto de la detección de malezas en campos de cultivos, varios estudios han aplicado la arquitectura U-Net con éxito. Por ejemplo, en el estudio de Kamilaris et al. (2018), se utilizó la red U-Net para segmentar y clasificar diferentes especies de malezas en imágenes de cultivos de maíz. El modelo entrenado demostró una alta precisión en la identificación de las malezas, lo que permitió tomar decisiones efectivas en el manejo de los cultivos.

De esta manera, la elección de la red U-Net como modelo principal en este proyecto se basa en su capacidad comprobada para la segmentación precisa de objetos en imágenes y su aplicabilidad en la detección de malezas en campos de cultivos. El lenguaje de programación para el desarrollo de esta CNN será Python utilizando librerías como TensorFlow, Keras y posteriormente entrenado en el entorno de Google Colaboratory en su versión de paga. Se utilizará la técnica de Grid Search para la etapa de entrenamiento (Syarif et al., 2016).

estadísticas y el uso de métricas establecidas por la norma ISO 25023. La Figura 3 se presenta la arquitectura de la aplicación móvil que se va a desarrollar en el trabajo de titulación.

Figura 3

Arquitectura de la aplicación móvil



Nota. Elaboración propia

Metodología

Para llevar a cabo con éxito el primer objetivo planteado en el proyecto se recopilará información de varios sitios en la red, como artículos científicos, trabajos de titulación, libros, entrevistas, conferencias, videos, etc., de los motores de bases bibliográficas que nos pone a disposición la Universidad Técnica del Norte. Dicha información deberá responder las preguntas planteadas y especificadas por el director del proyecto, además de la pregunta ¿Cómo se aplican las redes neuronales convolucionales para la identificación y detección de malezas en los cultivos?

El segundo objetivo se llevará a cabo empleando la metodología KDD (Timarán Pereira et al., 2016), la cual aporta una gran ayuda al momento de la recolección y preprocesamiento de imágenes obtenidas con el dron en los campos de cultivo.

En el tercer objetivo se seguirá con la continuación de la metodología KDD (Timarán Pereira et al., 2016), para el entrenamiento y la identificación de patrones de la red neuronal convolucional. Para luego desarrollar una aplicación móvil en Xamarin, usando los principios del manifiesto ágil ya definidos anteriormente. También, se hará uso de bibliotecas necesarias para la integración de modelos de aprendizaje automático y su ejecución en dispositivos

móviles. La red U-Net se caracteriza por su capacidad para capturar características a diferentes niveles de abstracción y su habilidad para generar segmentaciones precisas mediante la combinación de características de nivel local y global. La arquitectura U-Net ha sido ampliamente utilizada en diversas aplicaciones de segmentación, como la detección de células en imágenes microscópicas (Ronneberger et al., 2015) y la segmentación de órganos en imágenes médicas (Çiçek et al., 2016).

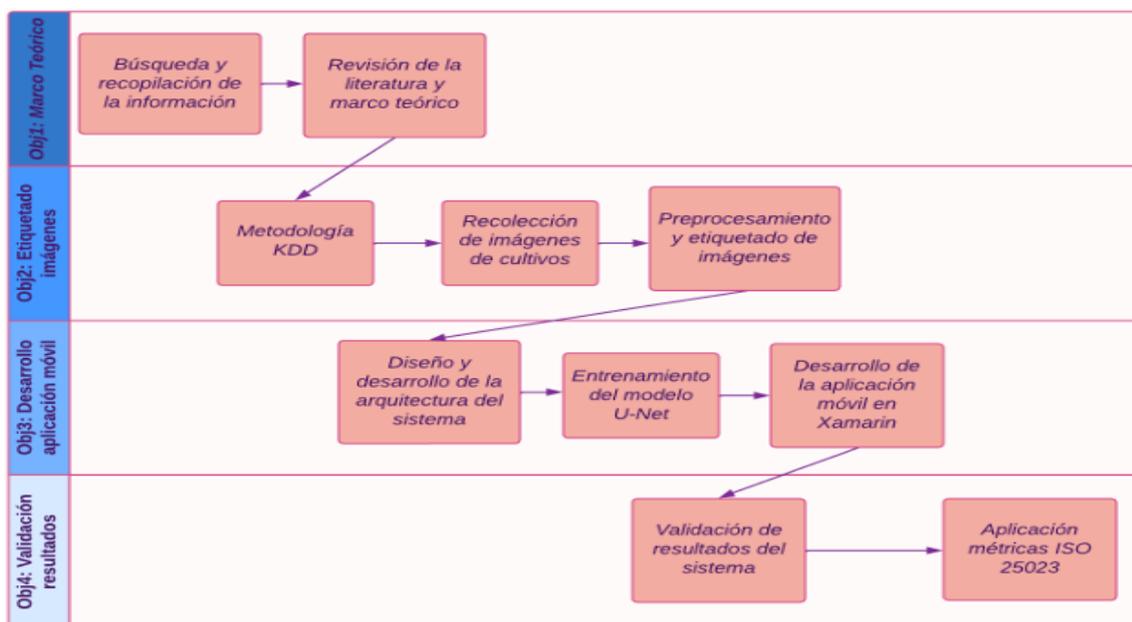
Finalmente, se evaluarán los resultados del sistema propuesto utilizando métricas estadísticas y métricas establecidas por la norma ISO 25023 para la evaluación de sistemas de software. Estas métricas permitirán medir la precisión y eficiencia del sistema en la identificación y cuantificación de malezas en los cultivos de maíz y/o papa (ISO, 2016).

Con este enfoque, se espera desarrollar un sistema completo y funcional que permita una detección precisa y eficiente de malezas en campos de cultivos de maíz y/o papa, mejorando así el manejo y control de estas en beneficio de los agricultores y el rendimiento de los cultivos.

La figura 4 muestra la metodología que seguirá el proyecto y los procesos que se llevarán a cabo para cumplir con todos los objetivos planteados.

Figura 4

Metodología



Nota. Elaboración propia

Justificación

El presente trabajo de titulación pretende responder y contribuir con el Objetivo de Desarrollo Sostenible Nro.12 “Producción y consumo responsables”, específicamente con la meta 12.2, la cual consiste en deducir la cantidad de pérdida de productos durante el proceso de siembra (Nation United, 2017).

Justificación Tecnológica

Se basa en el uso de tecnologías emergentes que han demostrado ser eficientes en aplicaciones similares como la combinación de tecnologías como el uso de drones, la arquitectura U-Net y técnicas de inteligencia artificial, que han demostrado ser efectivas en la detección y cuantificación automática de malezas en campos de cultivos. El uso de drones para la adquisición de imágenes en campos de cultivos ha sido ampliamente utilizado en la agricultura de precisión debido a su capacidad para capturar datos detallados y de alta resolución. Según Juhász et al. (2020), el uso de drones en la detección de malezas ha mostrado resultados prometedores en términos de precisión y eficiencia. La aplicación de técnicas de inteligencia artificial, como las redes neuronales convolucionales, en la detección y cuantificación automática de malezas ha sido abordada por diversos estudios. Por ejemplo, Nilsback y Zisserman (2008) utilizaron redes neuronales convolucionales para clasificar diferentes especies de malezas en imágenes de cultivos.

Justificación Agrícola

Se fundamenta en la necesidad de mejorar la identificación y cuantificación de malezas en los campos de cultivos de maíz y/o papa. La presencia de estas malas hierbas representa una amenaza significativa para el rendimiento y la calidad de la cosecha, ya que compiten por los recursos del suelo y pueden actuar como hospederos de plagas y enfermedades. La identificación y cuantificación manual de malezas es laboriosa, propensa a errores y consume tiempo y mano de obra. Mediante el uso de imágenes adquiridas por un dron y la aplicación de la red neuronal convolucional U-Net, se propone una solución automatizada que mejorará la precisión y eficiencia en este proceso, permitiendo implementar estrategias de manejo y control efectivas. Esto contribuirá a mejorar la productividad y rentabilidad de los cultivos agrícolas (Craine & Dybzinski, 2013; Pusd-Chulde et al., 2020; Smith et al., 2020).

Justificación Econmica

Radica en los beneficios que se derivarn de la implementacin de un sistema automatizado de identificacin y cuantificacin de malezas en campos de cultivos de maz y/o papa. La presencia de malezas representa una amenaza significativa para el rendimiento de los cultivos y la calidad de la cosecha, lo que puede resultar en prdidas econmicas para los agricultores. La identificacin manual de malezas es un proceso laborioso y propenso a errores, lo que implica un consumo de tiempo y mano de obra considerable, as como costos

adicionales asociados a inspecciones minuciosas y la necesidad de implementar estrategias de control y uso de herbicidas ineficientes. Al automatizar este proceso mediante el uso de imágenes adquiridas por un dron y la aplicación de una red neuronal convolucional U-Net, se espera mejorar la precisión y eficiencia en la identificación y cuantificación de malezas, lo que permitirá tomar decisiones más efectivas en el control de malezas, optimizar el uso de recursos y reducir los costos asociados. Esto contribuirá a mejorar la rentabilidad de los cultivos y la sostenibilidad económica de las operaciones agrícolas (Smith et al., 2019; Li et al., 2020).

Riesgos

Los riesgos del proyecto se detallan en la tabla 1.

Tabla 1

Riesgos del proyecto

Nro.	Riesgo	Solución
R1	Falta de disponibilidad o calidad insuficiente de las imágenes adquiridas por el dron.	Se realizará una cuidadosa planificación y coordinación en la adquisición de imágenes por parte del dron. Además, se llevará a cabo un proceso de control de calidad de las imágenes adquiridas, descartando aquellas de baja calidad o que no cumplan con los requisitos necesarios para el entrenamiento de la red neuronal.
R2	Limitaciones en el rendimiento y la eficiencia de la red neuronal convolucional U-Net.	Para abordar este riesgo, se realizará un exhaustivo proceso de entrenamiento de la red neuronal, utilizando conjuntos de datos de entrenamiento representativos y de alta calidad. Se optimizarán los hiperparámetros del modelo y se realizarán pruebas rigurosas para evaluar su rendimiento y eficiencia.
R3	Problemas de integración de la aplicación móvil con la red neuronal y las funcionalidades requeridas.	Se realizará un plan detallado de desarrollo de la aplicación móvil, definiendo claramente los requisitos de integración con la red neuronal y las funcionalidades específicas necesarias para la identificación y cuantificación de malezas.
R4	Desastres naturales	Tener la disponibilidad de otras plantaciones de maíz y/o papa
R5	Falta de compromiso por el administrador de la granja La Pradera de la UTN	Se buscará no depender únicamente de una autoridad de la plantación
R6	Falta de experiencia para el etiquetado de las imágenes	Adquirir conocimiento mediante fuentes de información
R7	Plantaciones de maíz y/o papa con mínima cantidad de malezas	Esperar que dichos cultivos presenten una mayor cantidad de malezas
R8	Inexistencia de cultivos de maíz y/o papa en la granja La Prader de la UTN	Tener la disponibilidad de otras plantaciones de maíz y/o papa
R9	Poco tiempo de participación del director, asesor y especialista del trabajo de titulación	Se buscará aprovechar las reuniones con el directo, asesor y especialista
R10	Malas condiciones climáticas	Aprovechar las buenas condiciones climáticas para la recolección de imágenes

Nota. Elaboración propia

Tabla 2

Matriz de riesgos

Nivel de riesgo	Alto		R10: Malas condiciones climáticas	R9: Poca participación de miembros del proyecto
	Medio		R2: Limitaciones de la CNN R6: Falta de experiencia R7: Maleza mínima	R1: No disponibilidad del dron R3: Problemas de integración
	Bajo	R8: Inexistencia de cultivos	R5: Falta de compromiso del administrador	R4: Desastres naturales
		Bajo	Medio	Alto
	Impacto de riesgo			

Nota. Elaboración propia

CAPÍTULO 1

Marco teórico

1.1 Contexto General

1.1.1 Agricultura y detección de malezas

La agricultura juega un papel fundamental en la producción de alimentos y la economía mundial. Sin embargo, enfrenta un importante desafío con la presencia de malezas en los campos de cultivo de maíz y papa, lo que afecta negativamente el rendimiento y la calidad de los cultivos (Bai et al., 2014; Garcia-Santillan & Pajares, 2018). Detectar y cuantificar las malezas de manera temprana resulta crucial para minimizar su impacto negativo en la agricultura. El control efectivo de malezas es esencial para garantizar la productividad y calidad de los cultivos. Estas plantas no deseadas compiten con los cultivos por nutrientes, luz y agua, lo que puede reducir significativamente los rendimientos agrícolas. Estudios indican que las malezas pueden disminuir los rendimientos entre un 10% y un 80%, dependiendo de varios factores (Adrian et al., 2005).

Históricamente, el control de malezas ha sido una labor costosa y laboriosa, empleando métodos tradicionales como el deshierbe manual y el uso de herbicidas químicos. Sin embargo, estos enfoques tienen desventajas, como la dependencia de mano de obra, el impacto negativo en el medio ambiente y el desarrollo de resistencia a herbicidas convencionales. En las últimas décadas, se ha avanzado significativamente en el uso de tecnologías basadas en inteligencia artificial y aprendizaje automático para detectar y controlar malezas. Mediante el uso de imágenes y datos recopilados por drones, cámaras agrícolas y sensores remotos, se han desarrollado sistemas automatizados para identificar y cuantificar malezas en campos de cultivo.

El análisis de imágenes agrícolas a través de técnicas de visión por computadora y algoritmos de aprendizaje profundo, como las redes neuronales convolucionales (CNN), ha demostrado ser una estrategia eficaz para la detección temprana y segmentación semántica de malezas en cultivos de maíz y papa (Jin et al., 2021). Las CNN tienen la capacidad de extraer características relevantes de las imágenes agrícolas, lo que les permite identificar y clasificar las malezas de manera precisa y automatizada.

Uno de los modelos de red neuronal convolucional más ampliamente utilizado para la segmentación semántica de malezas en imágenes agrícolas es U-Net. Esta arquitectura única combina un camino de codificación y un camino de decodificación, lo que permite capturar detalles finos y realizar predicciones precisas, mejorando significativamente la eficiencia y exactitud en la detección de malezas en comparación con los métodos tradicionales (Chen et al., 2023).

La adopción de tecnologías de detección de malezas basadas en inteligencia artificial

ofrece diversos beneficios. En primer lugar, proporciona una detección temprana y precisa de las malezas, permitiendo a los agricultores tomar decisiones oportunas sobre estrategias de control. En segundo lugar, reduce la dependencia de productos químicos agrícolas y minimiza el impacto ambiental negativo asociado al uso excesivo de herbicidas. En tercer lugar, la automatización de la detección de malezas a través de tecnologías avanzadas puede disminuir significativamente los costos laborales y aumentar la eficiencia en la producción agrícola.

A pesar de los avances en la detección de malezas basada en inteligencia artificial, es importante tener en cuenta que existen desafíos y limitaciones. Identificar especies de malezas específicas puede ser un reto, ya que algunas pueden parecerse a los cultivos u otras plantas. Además, contar con conjuntos de datos de alta calidad y representativos resulta crucial para el entrenamiento y ajuste óptimo de los modelos de detección de malezas (Sood et al., 2022).

La detección de malezas en la agricultura ha evolucionado significativamente gracias al avance de la inteligencia artificial y el aprendizaje automático. La aplicación de técnicas de visión por computadora y el uso de redes neuronales convolucionales, como U-Net, han demostrado ser herramientas poderosas para detectar de manera temprana y precisa las malezas en campos de cultivo de maíz y papa. La adopción de estas tecnologías ofrece múltiples ventajas para los agricultores, incluyendo la mejora de la productividad, la sostenibilidad ambiental y la reducción de costos operativos.

1.1.2 Tecnologías para la cuantificación automática de malezas

En el ámbito agrícola, se han utilizado diversos enfoques para la detección y cuantificación de malezas. Algunas tecnologías tradicionales se basan en técnicas de agrupación y modelado morfológico para segmentar la vegetación en imágenes agrícolas (Bai et al., 2013). Sin embargo, en la búsqueda de métodos más avanzados, el uso de imágenes adquiridas por drones ha demostrado ser una opción prometedora para la detección automática de malezas debido a su capacidad para abarcar grandes áreas y proporcionar información detallada (Romeo et al., 2013).

La cuantificación automática de malezas en campos de cultivo es una tarea crucial para mejorar la eficiencia y la productividad agrícola. La presencia de malezas puede afectar negativamente el rendimiento de los cultivos, compitiendo por los recursos esenciales y afectando la calidad de las cosechas. Por lo tanto, es de vital importancia contar con tecnologías avanzadas que permitan una detección precisa y automatizada de las malezas en los campos de maíz y papa, lo que proporciona una base sólida para el desarrollo de estrategias efectivas de manejo y control.

En la última década, se ha producido un avance significativo en el desarrollo de

tecnologías para la cuantificación automática de malezas, en gran medida gracias al uso de técnicas de inteligencia artificial y aprendizaje automático. Estas tecnologías utilizan imágenes adquiridas por drones, cámaras agrícolas y sensores remotos para capturar datos detallados de los campos de cultivo y, posteriormente, aplican algoritmos avanzados para identificar y segmentar las malezas con precisión.

La inteligencia artificial, en particular, ha desempeñado un papel fundamental en el desarrollo de estas tecnologías. Los algoritmos de aprendizaje profundo, como las redes neuronales convolucionales (CNN), han demostrado ser altamente eficaces en la detección y segmentación de objetos en imágenes, incluidas las malezas en campos de cultivo. Estos modelos de CNN, entrenados con conjuntos de datos adecuados y representativos, pueden aprender patrones complejos y características específicas de las malezas, lo que les permite realizar una identificación precisa incluso en condiciones variables de iluminación y en presencia de cultivos superpuestos (Jin et al., 2021).

Otra tecnología clave para la cuantificación automática de malezas es la visión por computadora. Esta disciplina se enfoca en el desarrollo de algoritmos y técnicas que permiten a las máquinas interpretar y comprender imágenes digitales. En el contexto de la agricultura, la visión por computadora se ha aplicado con éxito para la identificación y clasificación de malezas, mediante la extracción de características relevantes y la utilización de métodos de segmentación semántica para separar las malezas de los cultivos circundantes (Bai et al., 2014).

Además de las tecnologías de inteligencia artificial y visión por computadora, también se han empleado técnicas de procesamiento de imágenes y análisis espectral para la cuantificación automática de malezas. El uso de imágenes multiespectrales y térmicas ha permitido detectar diferencias sutiles en la reflectancia y la temperatura de las malezas y los cultivos, lo que contribuye a una identificación más precisa y a la evaluación del estado de salud de los cultivos (García-Santillán & Pajares, 2018).

Es importante mencionar que el desarrollo de tecnologías para la cuantificación automática de malezas se ha beneficiado significativamente de la disponibilidad de grandes conjuntos de datos y el acceso a recursos informáticos avanzados. La cantidad y calidad de los datos de entrenamiento son fundamentales para el éxito de los modelos de inteligencia artificial y aprendizaje automático, ya que permiten a los algoritmos aprender patrones y características relevantes para una detección precisa (Lu et al., 2022).

Las tecnologías para la cuantificación automática de malezas en campos de cultivo de maíz y papa han experimentado un progreso significativo gracias a la aplicación de técnicas de inteligencia artificial, aprendizaje automático, visión por computadora y análisis espectral. Estas tecnologías ofrecen una solución prometedora para la detección temprana y precisa de malezas, lo que puede mejorar la eficiencia agrícola, reducir los costos de manejo y promover

prácticas agrícolas más sostenibles y respetuosas con el medio ambiente.

1.2 Fundamentos de las redes neuronales convolucionales (CNN)

1.2.1 Introducción a las redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son una clase de algoritmos de aprendizaje profundo ampliamente utilizados en la visión por computadora y otras áreas. Estas redes están diseñadas para procesar imágenes y reconocer patrones complejos mediante el uso de capas convolucionales y de agrupación (Jin et al., 2021). Su estructura jerárquica les permite aprender características discriminativas de forma automática y realizar tareas de segmentación semántica con alta precisión.

Las Redes Neuronales Convolucionales (CNN por sus siglas en inglés) han surgido como una de las arquitecturas más poderosas y revolucionarias dentro del campo del aprendizaje profundo y la visión por computadora (LeCun et al., 1998; Krizhevsky et al., 2012). Estas redes están inspiradas en el funcionamiento del sistema visual biológico y han demostrado ser altamente efectivas en el procesamiento y análisis de imágenes, superando significativamente a los métodos tradicionales en tareas como reconocimiento de objetos, clasificación y segmentación semántica (Garcia-Santillan & Pajares, 2018; Chen et al., 2023).

El concepto central que impulsa el éxito de las CNN es la convolución, que se basa en la idea de aplicar filtros o kernels a la imagen para detectar características locales y patrones significativos (LeCun et al., 1998). La convolución permite que la red aprenda y extraiga automáticamente características relevantes, como bordes, texturas y formas, a través del entrenamiento con grandes conjuntos de datos etiquetados (Krizhevsky et al., 2012). Esta habilidad para identificar características jerárquicas en diferentes niveles de abstracción hace que las CNN sean extremadamente adecuadas para la representación y comprensión de imágenes complejas.

El diseño típico de una CNN consta de múltiples capas, incluyendo capas de convolución, capas de activación (como ReLU), capas de pooling y capas completamente conectadas (LeCun et al., 1998). Las capas de convolución se encargan de realizar la convolución y extraer características, mientras que las capas de pooling reducen la dimensionalidad de la representación espacial y ayudan a obtener características invariantes a la traslación (Krizhevsky et al., 2012). Estas capas, junto con las funciones de activación, permiten que la red aprenda de manera no lineal y capturar patrones más complejos.

El proceso de entrenamiento de una CNN implica la optimización iterativa de los pesos y parámetros de la red para minimizar una función de pérdida que mide la discrepancia entre las predicciones del modelo y las etiquetas reales del conjunto de entrenamiento (LeCun et al., 1998). Para llevar a cabo esta optimización, se utilizan algoritmos de optimización como el descenso de gradiente estocástico (SGD) o variantes mejoradas como el optimizador Adam

(Kingma & Ba, 2015). A través de este proceso, la red ajusta sus parámetros para mejorar su capacidad de generalización y realizar predicciones precisas en datos no vistos.

Las CNN han demostrado su valía en una amplia gama de aplicaciones prácticas, desde la detección de objetos en imágenes (García-Santillán & Pajares, 2018) hasta la segmentación de objetos y regiones de interés (Chen et al., 2023). También han encontrado aplicación en campos como la medicina, donde se utilizan para tareas de diagnóstico y análisis de imágenes médicas (Liu et al., 2023), y en la agricultura, donde se emplean para la detección de malezas (Jin et al., 2021) y enfermedades en cultivos (Panchal et al., 2023).

En general, las Redes Neuronales Convolucionales han demostrado ser una tecnología revolucionaria en el campo de la visión por computadora y el procesamiento de imágenes. Su capacidad para aprender y extraer características de manera automática y jerárquica, junto con su versatilidad en diversas aplicaciones, ha impulsado importantes avances en áreas como la inteligencia artificial, la medicina y la agricultura.

1.2.2 Redes neuronales convolucionales en la detección de malezas

En la detección de malezas en campos de cultivo, las redes neuronales convolucionales han demostrado ser eficaces en la segmentación semántica de objetos vegetales (Chen et al., 2023). Estas redes pueden distinguir entre malezas y cultivos mediante el análisis de patrones visuales en las imágenes agrícolas. El uso de técnicas avanzadas de deep learning, como U-Net, ha mejorado significativamente la precisión y eficiencia de la detección de malezas en comparación con métodos tradicionales (Chakraborty et al., 2022).

Las Redes Neuronales Convolucionales han demostrado ser una herramienta poderosa y eficiente para abordar el desafío de la detección de malezas en campos de cultivos de maíz y/o papa. La detección precisa y oportuna de malezas es de vital importancia para la agricultura moderna, ya que las malezas pueden competir con los cultivos por recursos como nutrientes y agua, lo que resulta en una reducción significativa en el rendimiento y la calidad de las cosechas (Sparrow et al., 2021; Lu et al., 2022).

Las CNN son una subclase de las redes neuronales artificiales que se caracterizan por su capacidad para aprender patrones y características relevantes en imágenes de manera automática. Esta habilidad ha llevado a un aumento significativo en la precisión y eficiencia de la detección de objetos en imágenes. En el contexto de la detección de malezas, las CNN son entrenadas para identificar y segmentar regiones que corresponden a las malezas en las imágenes capturadas en campos de cultivos (Dudewicz, 1989; Chen et al., 2023).

El proceso de detección de malezas utilizando CNN comienza con la adquisición de imágenes de alta resolución de los campos mediante diversas tecnologías, como drones equipados con cámaras RGB o multiespectrales (Romeo et al., 2013). Estas imágenes son preprocesadas para mejorar la calidad y resolución, y luego se utilizan como entrada para la

CNN.

La CNN consta de múltiples capas, incluidas capas de convolución y pooling, que se encargan de extraer características relevantes de las imágenes. Estas capas aprenden filtros y patrones específicos durante el proceso de entrenamiento, lo que permite que la red identifique patrones distintivos asociados con las malezas. Además, las capas completamente conectadas y las funciones de activación, como ReLU, ayudan a la CNN a realizar la clasificación y segmentación de las malezas en las imágenes (Chakraborty et al., 2022; Zhang & Zhang, 2023).

El entrenamiento de la CNN se realiza utilizando un conjunto de datos etiquetado que contiene imágenes de campos de cultivos junto con las correspondientes máscaras de segmentación que indican la ubicación de las malezas. A través del proceso de optimización iterativa, la CNN ajusta sus parámetros para minimizar la función de pérdida y mejorar la precisión de la detección de malezas en imágenes no vistas anteriormente (Bai et al., 2014).

Una vez que la CNN ha sido entrenada, se puede utilizar para la detección de malezas en nuevas imágenes. La red toma una imagen como entrada y produce un mapa de probabilidad que indica la presencia y ubicación de las malezas en la imagen. Posteriormente, se pueden aplicar técnicas de posprocesamiento para eliminar falsos positivos y mejorar la precisión general de la detección (Jin et al., 2021).

El uso de CNN en la detección de malezas ha demostrado ser altamente efectivo y ha superado a los enfoques tradicionales basados en métodos de umbral y segmentación de color (Bai et al., 2013). La capacidad de las CNN para aprender características discriminativas de manera automática y su capacidad para generalizar a nuevas imágenes hacen que sean una herramienta valiosa para los agricultores en la lucha contra las malezas y la mejora de la eficiencia de la producción agrícola (Shaikh et al., 2022; Garcia-Santillan & Pajares, 2018).

En conclusión, las Redes Neuronales Convolucionales han demostrado ser una tecnología prometedora para la detección de malezas en campos de cultivos de maíz y/o papa. Su capacidad para aprender patrones y características relevantes en imágenes, combinada con su alta precisión y eficiencia, ha revolucionado la forma en que se aborda el desafío de las malezas en la agricultura moderna.

1.3 La red neuronal convolucional U-Net

1.3.1 Introducción a U-Net

U-Net es una arquitectura especializada de red neuronal convolucional que ha sido diseñada específicamente para tareas de segmentación semántica (Chen et al., 2023). Su diseño único de codificación y decodificación permite capturar detalles finos en las imágenes y realizar predicciones precisas. El modelo U-Net ha sido ampliamente utilizado en aplicaciones de visión por computadora, incluida la detección de malezas en campos de cultivo

de maíz y papa (Shaikh et al., 2022).

La arquitectura de la red neuronal convolucional U-Net, propuesta por Olaf Ronneberger, Philipp Fischer y Thomas Brox en 2015, ha revolucionado la tarea de segmentación semántica en imágenes médicas y de visión por computadora. U-Net es ampliamente conocida por su eficacia en la segmentación precisa de objetos, incluso en escenarios con conjuntos de datos pequeños y desequilibrados. Su diseño único, que incorpora una estructura de encoder-decoder con conexiones skip, permite lograr resultados sobresalientes en tareas de segmentación, incluida la cuantificación automática de malezas en campos de cultivos de maíz y/o papa a partir de imágenes adquiridas por un dron.

El nombre "U-Net" proviene de la forma de su arquitectura, que se asemeja a la letra "U". La red consiste en dos partes principales: el encoder y el decoder. El encoder está compuesto por múltiples capas de convolución y max pooling, que se encargan de extraer características de nivel superior de la imagen y reducir su tamaño espacial. Esta parte de la red actúa como un extractor de características y ayuda a capturar información relevante y significativa en la imagen, lo que facilita la tarea de segmentación (Ronneberger et al., 2015; Romeo et al., 2013).

En contraste, el decoder es responsable de realizar la segmentación precisa y detallada. Utiliza capas de upsampling y convolución transpuesta para aumentar la resolución de las características extraídas por el encoder. Además, U-Net utiliza conexiones skip que conectan las capas del encoder con las correspondientes del decoder. Estas conexiones permiten que la información detallada se transmita desde las capas del encoder directamente a las capas del decoder, lo que ayuda a preservar los detalles de la imagen original durante el proceso de segmentación (Chen et al., 2023).

La arquitectura U-Net ha sido ampliamente adoptada en diversas aplicaciones, incluida la cuantificación automática de malezas en la agricultura. Su eficacia en la segmentación semántica y su capacidad para adaptarse a conjuntos de datos pequeños han demostrado ser altamente beneficiosas en la detección y eliminación de malezas en campos de cultivos de maíz y/o papa. Al entrenar la red con imágenes de campos agrícolas y sus respectivas máscaras de segmentación que indican la ubicación de las malezas, U-Net puede aprender a identificar y delinear con precisión las áreas afectadas por malezas (Adrian et al., 2005; Panchal et al., 2023).

La utilización de U-Net en la detección de malezas ha llevado a mejoras significativas en la eficiencia y la productividad en la agricultura. Al permitir la detección temprana y precisa de malezas, los agricultores pueden implementar medidas de control específicas y focalizadas, reduciendo así la competencia por recursos con los cultivos de maíz y/o papa y mejorando el rendimiento general de las cosechas. Además, la aplicación de U-Net en el análisis de imágenes obtenidas por drones ha permitido una cobertura más amplia y una

inspección más detallada de los campos agrícolas, lo que resulta en una gestión más efectiva y sostenible de las malezas (Chakraborty et al., 2022; Zhang et al., 2023).

En resumen, la arquitectura U-Net ha demostrado ser una herramienta esencial en la cuantificación automática de malezas en campos de cultivos de maíz y/o papa a través del análisis de imágenes capturadas por drones. Su diseño único y sus conexiones skip han permitido obtener resultados sobresalientes en tareas de segmentación semántica, proporcionando a los agricultores una valiosa herramienta para combatir las malezas y mejorar la eficiencia de la producción agrícola.

1.3.2 Arquitectura y funcionamiento de U-Net

La arquitectura de U-Net consta de capas convolucionales que forman un camino de codificación y un camino de decodificación. El camino de codificación extrae características relevantes de la imagen, mientras que el camino de decodificación reconstruye la imagen segmentada (Chen et al., 2023). La estructura en forma de U de U-Net permite una fusión efectiva de características, lo que resulta en una segmentación semántica precisa y detallada de las malezas en las imágenes agrícolas (Liu et al., 2023).

La arquitectura U-Net es un enfoque novedoso y altamente efectivo en el campo de la visión por computadora y el procesamiento de imágenes médicas. Fue propuesta por Ronneberger et al. (2015) como una solución para el desafío de la segmentación de imágenes biomédicas con una cantidad limitada de datos de entrenamiento. U-Net se ha convertido en una herramienta fundamental en aplicaciones como la detección de objetos, la segmentación semántica y el análisis de imágenes médicas.

La estructura de U-Net se caracteriza por su forma simétrica en forma de U, de ahí su nombre. Esta arquitectura está formada por dos partes principales: el camino de contracción y el camino de expansión. El camino de contracción consiste en una serie de capas convolucionales seguidas de capas de pooling, que trabajan en conjunto para capturar y resumir la información relevante de la imagen de entrada. Este proceso reduce gradualmente la resolución espacial de la imagen, lo que conduce a la extracción de características de alto nivel.

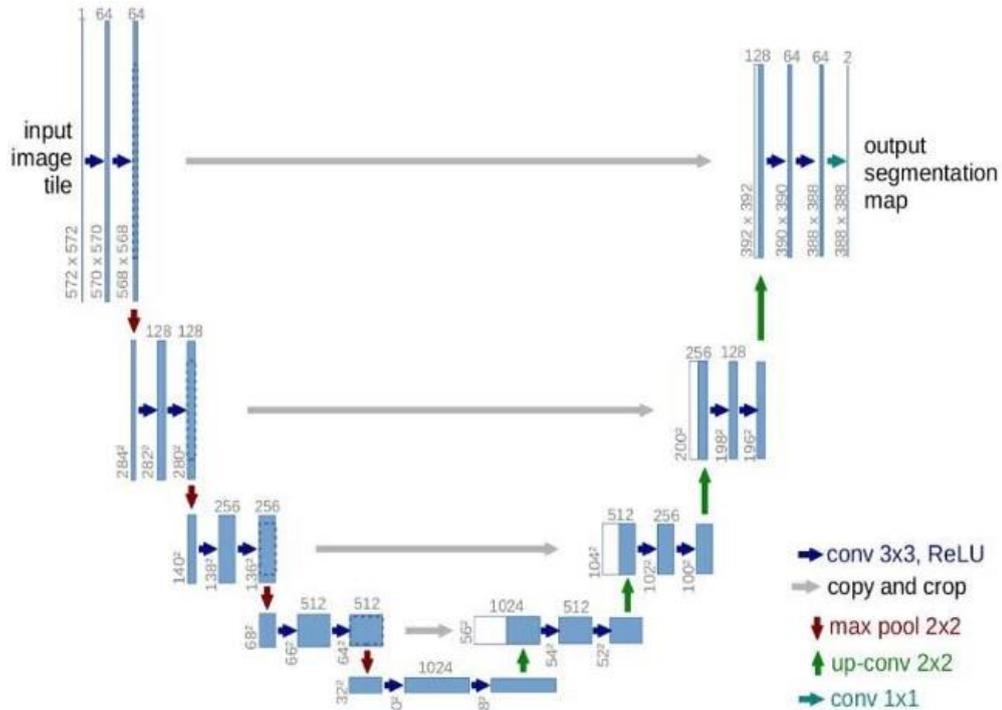
Por otro lado, el camino de expansión realiza la tarea inversa mediante la aplicación de operaciones de up-sampling y concatenación de características para lograr una reconstrucción precisa y detallada de la imagen segmentada. La combinación de estas dos partes permite que U-Net tenga una arquitectura de codificador-decodificador, lo que facilita la localización precisa de objetos de interés en la imagen.

El corazón de U-Net son los bloques de convolución, que se utilizan para extraer características y representaciones de la imagen de entrada. Estos bloques generalmente constan de múltiples capas convolucionales, seguidas de funciones de activación, como ReLU

(Rectified Linear Unit), que ayudan a introducir no linealidades en el modelo. Además, para reducir el riesgo de overfitting y mejorar la generalización, U-Net suele incorporar técnicas de regularización, como la eliminación aleatoria de nodos o la normalización por lotes.

Figura 2

Arquitectura de la red neuronal convolucional U-NET.



Nota. (Ronneberger et al., 2015)

U-Net ha demostrado su eficacia en una variedad de tareas de segmentación de imágenes, destacando especialmente en aplicaciones biomédicas, como la segmentación de células y órganos en imágenes de microscopía o resonancias magnéticas. La naturaleza de su arquitectura permite que U-Net se adapte a conjuntos de datos pequeños y escasos, lo que la convierte en una opción atractiva cuando la disponibilidad de datos de entrenamiento es limitada.

Desde su presentación, U-Net ha sido objeto de diversas mejoras y extensiones. Varias variantes como U-Net++ (Zhou et al., 2018), Attention U-Net (Schlemper et al., 2019) y U-Net 3+ (Li et al., 2018) han sido propuestas para abordar limitaciones específicas o mejorar el rendimiento en determinadas tareas. Además, la incorporación de mecanismos de atención ha permitido que U-Net focalice mejor las regiones relevantes de la imagen, lo que ha llevado a una mayor precisión en la segmentación.

Entonces podemos decir que U-Net es una arquitectura profunda y versátil utilizada para la segmentación de imágenes que ha demostrado un excelente desempeño en diversas aplicaciones, particularmente en el campo de la visión por computadora y el procesamiento

de imágenes médicas.

1.3.3 Entrenamiento y ajuste de U-Net

Para utilizar U-Net en la detección de malezas, es necesario entrenar el modelo con un conjunto de datos adecuado (Chakraborty et al., 2022). Se deben considerar diferentes estrategias para el entrenamiento y ajuste de los hiperparámetros para obtener el mejor rendimiento posible en la segmentación de malezas en imágenes agrícolas (Chen et al., 2023).

El entrenamiento y ajuste de la red neuronal convolucional U-Net es un aspecto fundamental en la aplicación de esta arquitectura en tareas de segmentación semántica y detección de objetos en imágenes. Para lograr un rendimiento óptimo, se requiere un proceso de entrenamiento cuidadoso y un ajuste adecuado de los hiperparámetros. En este contexto, la presente sección aborda los aspectos clave relacionados con el entrenamiento de U-Net, incluidos los conjuntos de datos utilizados, la función de pérdida, el optimizador y las estrategias de ajuste.

El primer paso en el entrenamiento de U-Net es la preparación del conjunto de datos de entrenamiento. Este conjunto debe consistir en imágenes etiquetadas junto con sus respectivas máscaras de segmentación que indican la ubicación y los límites de las regiones de interés, como las malezas en la agricultura o las células en imágenes médicas. La cantidad y la calidad de los datos de entrenamiento son factores críticos que pueden afectar significativamente el rendimiento de la red. Es importante contar con un conjunto de datos lo suficientemente grande y representativo para garantizar que la red adquiera una comprensión generalizada de las características y variaciones de los objetos de interés (Zhang et al., 2018; Ronneberger et al., 2015).

Una vez que se ha preparado el conjunto de datos, se procede a definir la función de pérdida que guiará el proceso de entrenamiento. La elección de una función de pérdida adecuada es crucial para la convergencia eficiente de la red y la obtención de resultados precisos. Para problemas de segmentación, una función de pérdida comúnmente utilizada es la "binary cross-entropy" o entropía cruzada binaria, que mide la discrepancia entre las predicciones de la red y las máscaras de segmentación reales. La entropía cruzada binaria es particularmente adecuada para tareas de clasificación de píxeles, como la detección de malezas, ya que penaliza las predicciones erróneas de manera efectiva (Falk et al., 2019; Oktay et al., 2018).

Además de la función de pérdida, el proceso de entrenamiento también implica la elección de un optimizador que ajuste los pesos de la red para minimizar la función de pérdida. El optimizador más utilizado en el entrenamiento de U-Net es el "Adam" (Adaptive Moment Estimation), que combina el algoritmo de descenso de gradiente estocástico con tasas de aprendizaje adaptativas para una convergencia más rápida y estable. El optimizador Adam ha

demostrado ser eficiente en la mayoría de los problemas de entrenamiento de redes neuronales y es ampliamente utilizado en aplicaciones de U-Net (Kingma & Ba, 2017; Ghafoorian et al., 2017).

Otra consideración importante durante el entrenamiento es el ajuste de los hiperparámetros de la red, como la tasa de aprendizaje, el tamaño del lote y el número de épocas. La tasa de aprendizaje determina qué tan rápido la red aprende de los datos y puede afectar la velocidad y la estabilidad de la convergencia. Un tamaño de lote adecuado es esencial para garantizar un buen equilibrio entre el rendimiento y la eficiencia computacional. El número de épocas se refiere a la cantidad de veces que el conjunto de entrenamiento se pasa por la red durante el proceso de entrenamiento. Un número insuficiente de épocas puede dar lugar a un aprendizaje insuficiente, mientras que un número excesivo puede conducir al sobreajuste del modelo (Oktay et al., 2018; Falk et al., 2019).

El proceso de entrenamiento de U-Net implica el ajuste iterativo de los pesos de la red utilizando el conjunto de entrenamiento. Durante cada iteración, la red produce predicciones sobre el conjunto de entrenamiento y compara estas predicciones con las máscaras de segmentación reales utilizando la función de pérdida. Luego, el optimizador ajusta los pesos para minimizar la función de pérdida, lo que mejora gradualmente la capacidad de la red para realizar una segmentación precisa de los objetos de interés. La convergencia del entrenamiento se determina por criterios como la estabilidad de la función de pérdida y el rendimiento del conjunto de validación (Zhang et al., 2018; Falk et al., 2019).

Es importante destacar que el entrenamiento de U-Net puede ser computacionalmente intensivo y requerir un hardware adecuado, como unidades de procesamiento gráfico (GPU), para acelerar el proceso. La complejidad de la red y el tamaño del conjunto de datos también influyen en el tiempo de entrenamiento requerido. Sin embargo, una vez que el entrenamiento ha concluido, la red U-Net puede utilizarse para realizar inferencias rápidas y precisas en nuevas imágenes.

El entrenamiento y ajuste de la red U-Net son procesos cruciales para lograr una segmentación semántica efectiva en imágenes. La elección de un conjunto de datos representativo, una función de pérdida adecuada y un optimizador eficiente son aspectos esenciales para el éxito del entrenamiento. El ajuste adecuado de los hiperparámetros garantiza una convergencia estable y un rendimiento óptimo del modelo entrenado.

1.4 Metodología KDD

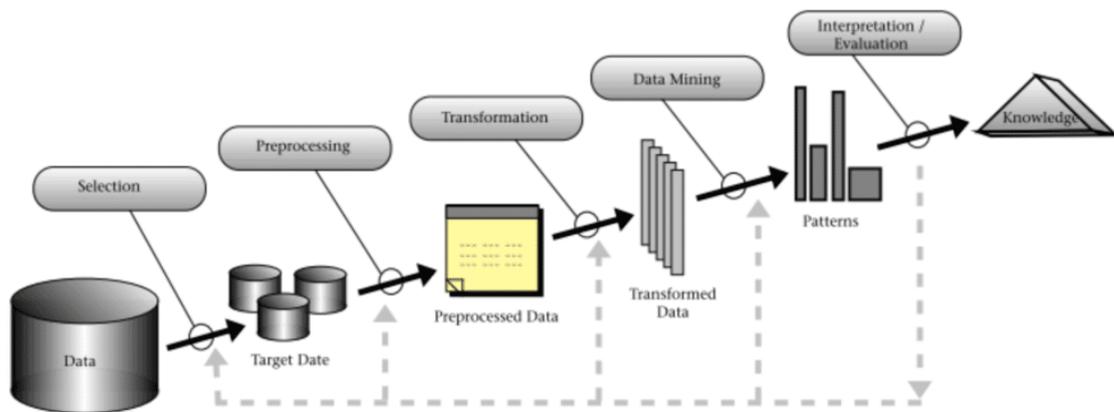
La Metodología KDD (Knowledge Discovery in Databases) es un enfoque sistemático y estructurado para descubrir conocimiento útil, novedoso y potencialmente válido a partir de grandes volúmenes de datos (Fayyad et al., 1996). Es una disciplina ampliamente utilizada en la minería de datos y el análisis de datos, que se basa en una serie de pasos secuenciales

para transformar datos sin procesar en información significativa y conocimiento útil. La Metodología KDD abarca desde la selección y preparación de los datos hasta la interpretación y evaluación de los resultados obtenidos, proporcionando una base sólida para la toma de decisiones informada y el descubrimiento de patrones ocultos.

En la figura 5 se muestra el proceso que sigue la metodología KDD.

Figura 5

Metodología KDD (Knowledge Discovery in Databases)



Nota. (Fayyad, U., 1996)

El primer paso en la Metodología KDD es la selección de los datos adecuados para el proceso de descubrimiento de conocimiento. Esto implica la identificación de las fuentes de datos relevantes, la definición de las variables de interés y la recopilación de los datos en una base de datos centralizada. Es fundamental garantizar la calidad y la integridad de los datos desde el inicio, ya que cualquier deficiencia en esta etapa puede tener un impacto significativo en la calidad de los resultados finales (Han et al., 2011).

Una vez que los datos han sido seleccionados y recopilados, el siguiente paso en la Metodología KDD es la limpieza y preprocesamiento de los datos. Esto implica la identificación y corrección de valores atípicos, datos faltantes y errores, así como la normalización y transformación de los datos para que estén en un formato adecuado para el análisis. Esta etapa es crucial para garantizar que los datos sean coherentes y confiables, lo que a su vez mejora la precisión y la confiabilidad de los resultados obtenidos (Fayyad et al., 1996).

Una vez que los datos han sido preparados, se procede a la etapa de minería de datos, que es el núcleo de la Metodología KDD. En esta etapa, se aplican diversas técnicas y algoritmos de análisis de datos para descubrir patrones, relaciones y tendencias ocultas en los datos. Estas técnicas pueden incluir algoritmos de clasificación, regresión, agrupamiento, asociación y otras técnicas avanzadas de aprendizaje automático y minería de datos (Witten

et al., 2011).

Una vez que se han obtenido los resultados de la minería de datos, se procede a la etapa de evaluación e interpretación de los resultados. En esta etapa, se analizan y validan los patrones y conocimientos descubiertos para asegurarse de que sean significativos, relevantes y útiles. Además, se evalúa la precisión y el rendimiento de los modelos y algoritmos utilizados para garantizar su eficacia en la solución del problema planteado (Fayyad et al., 1996).

Finalmente, la última etapa de la Metodología KDD es la presentación y aplicación del conocimiento descubierto. Los resultados y patrones identificados se comunican a los usuarios y partes interesadas relevantes en un formato comprensible y útil. Además, el conocimiento descubierto puede ser utilizado para tomar decisiones informadas, resolver problemas y mejorar procesos en diversas áreas, como la toma de decisiones empresariales, la medicina, la agricultura y muchas otras aplicaciones (Han et al., 2011).

En el contexto de la detección de malezas, la metodología KDD se puede aplicar para preprocesar y preparar los datos, entrenar y ajustar el modelo U-Net, y evaluar y validar los resultados obtenidos (Panchal et al., 2023).

En retrospectiva, la Metodología KDD es un enfoque estructurado y sistemático para descubrir conocimiento valioso a partir de grandes volúmenes de datos. Comienza con la selección y preparación de los datos, seguida de la limpieza y preprocesamiento de estos. Luego, se aplica la minería de datos para descubrir patrones y relaciones ocultas, seguida de la evaluación e interpretación de los resultados. Finalmente, el conocimiento descubierto se presenta y se aplica para tomar decisiones informadas y mejorar procesos en diversas áreas de aplicación.

1.5 Evaluación de la arquitectura de U-Net

1.5.1 Métricas de evaluación para la segmentación semántica

La evaluación precisa y objetiva de la segmentación semántica es esencial para medir el rendimiento y la eficacia de los algoritmos utilizados en esta tarea. Existen diversas métricas de evaluación que permiten cuantificar la calidad de la segmentación y comparar diferentes enfoques de manera justa y rigurosa. Estas métricas se utilizan para medir la concordancia entre las máscaras de segmentación producidas por el modelo y las máscaras de referencia, que representan la verdad terreno o ground-truth (García-Santillán & Pajares, 2018).

La evaluación del rendimiento de U-Net en la detección de malezas se realiza mediante el cálculo de diversas métricas de evaluación para la segmentación semántica. Entre las métricas comunes se encuentran la precisión, exhaustividad, F1-score y otras, que permiten comparar los resultados obtenidos con otros enfoques y analizar el desempeño del modelo

propuesto (Mishra et al., 2021).

Una de las métricas más comunes para la evaluación de la segmentación semántica es la precisión global o exactitud (accuracy), que mide la proporción de píxeles correctamente clasificados en relación con el total de píxeles en la imagen. Aunque la precisión global proporciona una visión general del rendimiento del modelo, puede no ser suficiente para evaluar la segmentación en casos donde hay desequilibrios de clases o clases con pocos píxeles (Chen et al., 2023). Por lo tanto, se utilizan otras métricas más específicas.

Otra métrica ampliamente utilizada es el coeficiente de Dice (Dice coefficient), también conocido como coeficiente de similitud de Sørensen-Dice. Esta métrica calcula el grado de superposición entre la máscara de segmentación y la máscara de referencia, proporcionando un valor entre 0 y 1, donde 1 indica una segmentación perfecta. El coeficiente de Dice es especialmente útil cuando se tienen clases desbalanceadas, ya que penaliza menos los falsos negativos y falsos positivos en comparación con la precisión global (Chen et al., 2023).

Además del coeficiente de Dice, la sensibilidad (recall) y la precisión (precision) son métricas importantes en la evaluación de la segmentación semántica. La sensibilidad mide la proporción de píxeles positivos correctamente identificados con respecto al total de píxeles positivos en la máscara de referencia. Por otro lado, la precisión mide la proporción de píxeles positivos correctamente identificados en la máscara de referencia en relación con el total de píxeles positivos en la máscara de segmentación. Estas métricas permiten evaluar la capacidad del modelo para detectar correctamente la presencia de cada clase de interés y cuantificar el nivel de precisión en sus predicciones (Chen et al., 2023).

Además de las métricas mencionadas anteriormente, existen otras métricas más avanzadas que se utilizan en la evaluación de la segmentación semántica, como la especificidad (specificity), el índice Jaccard (Jaccard index), el coeficiente de correlación de Matthews (Matthews correlation coefficient) y el índice de Youden (Youden's index). Estas métricas proporcionan una visión más completa y detallada del rendimiento del modelo y son especialmente útiles cuando se necesitan tener en cuenta factores como la sensibilidad a falsos positivos o falsos negativos en aplicaciones específicas (Chen et al., 2023).

La evaluación de la segmentación semántica es fundamental para medir la calidad y el rendimiento de los algoritmos utilizados en esta tarea. Las métricas de evaluación, como la precisión global, el coeficiente de Dice, la sensibilidad y la precisión, permiten cuantificar el grado de concordancia entre las máscaras de segmentación y las máscaras de referencia, brindando una visión objetiva del rendimiento del modelo. Además, existen otras métricas más avanzadas que proporcionan una evaluación más detallada y precisa del rendimiento, adaptándose a las necesidades específicas de cada aplicación.

TRABAJOS RELACIONADOS

Los trabajos relacionados con el proyecto de investigación se muestran en la tabla 3.

Tabla 3

Trabajos relacionados

Contexto internacional en base a tesis, trabajos o investigaciones realizadas:		
INVESTIGACION	ENLACE	APORTE
An Improved U-Net Model Based on Multi-Scale Input and Attention Mechanism: Application for Recognition of Chinese Cabbage and Weed (Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	https://www.scopus.com/inward/record.uri?eid=2-s2.0-85152788818&doi=10.3390%2fsu15075764&partnerID=40&md5=5350814ee5fa0e3c46cdffb892988296	Este estudio propone un método de segmentación semántica, MSECA-Unet, basado en una mejora de U-Net, para la identificación eficiente y precisa de cultivos de repollo chino y malezas en el campo, lo cual es crucial para realizar rociado preciso de herbicidas y operaciones de desmalezado inteligente. La mejora consiste en la integración lateral de entradas multi-escalares y la introducción de un mecanismo de atención de canal eficiente (ECA), reduciendo significativamente el número de parámetros y el tamaño del modelo. Los resultados experimentales muestran que el modelo propuesto supera a los modelos de segmentación semántica populares, PSPNet y DeepLab V3+, así como al modelo U-Net original, con un aumento en la intersección media sobre unión (MIOU) y precisión media de píxeles (MPA). Además, el modelo propuesto es más ligero y tiene una velocidad de reconocimiento rápida, pero tiene limitaciones en la identificación de especies de malezas y en su adaptabilidad a diferentes cultivos. Se sugiere realizar investigaciones futuras para mejorar la clasificación de especies de malezas y expandir el conjunto de datos para entrenar modelos más adaptables.
Deep learning-based crop row detection for infield navigation of agri-robots (de Silva R., Cielniak G., Wang G., Gao J., 2023)	https://www.scopus.com/inward/record.uri?eid=2-s2.0-85166915618&doi=10.1002%2frob.22238&partnerID=40&md5=571d3f4cb10fd53d5426b87b60a0ff5e	El estudio presenta un algoritmo robusto de detección de hileras de cultivos que puede resistir las variaciones en el campo utilizando cámaras económicas. Se creó un conjunto de datos de imágenes de remolacha azucarera que representa 11 variaciones en el campo, incluyendo diferentes etapas de crecimiento, niveles de luz, densidades de malezas variables, hileras de cultivos curvas y discontinuas. El algoritmo propuesto segmenta las hileras de cultivos utilizando un método basado en aprendizaje profundo y emplea la máscara

		<p>de segmentación predicha para la extracción del cultivo central mediante un novedoso algoritmo de selección de hileras centrales de cultivo. Se probó el algoritmo en la detección de hileras de cultivos y en la capacidad de navegación visual a lo largo de una hilera de cultivos, obteniendo un rendimiento robusto en condiciones desafiantes del campo. La simulación en Gazebo demostró que el método puede seguir las hileras de cultivos con precisión, mejorando significativamente respecto al enfoque de referencia.</p>
<p>Improving U-net network for semantic segmentation of corns and weeds during corn seedling stage in field (Cui J, Tan F, Bai N and Fu Y, 2024)</p>	<p>https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2024.1344958/full</p>	<p>El estudio propone una red de segmentación semántica llamada RDS_Unet para identificar malezas en campos de plántulas de maíz. Esta red utiliza mejoras en el modelo U-net, incorporando características de ResNeXt-50 y convoluciones deformables. Los resultados muestran un rendimiento superior en comparación con modelos clásicos, lo que sugiere su eficacia en equipos de control mecánico de malezas.</p>
<p>Modified U-Net for plant diseased leaf image segmentation (Shanwen Zhang, Chuanlei Zhang, 2023)</p>	<p>https://doi.org/10.1016/j.compag.2022.107511</p>	<p>Este estudio presenta un método basado en una mejora de U-Net (MU-Net) para la segmentación de imágenes de hojas de plantas enfermas. Se introduce un bloque residual (Resblock) y un camino residual (Respath) para mejorar la profundidad de la red y su capacidad de expresión. Los resultados experimentales muestran que este método mejora la precisión y eficiencia de la segmentación de imágenes de hojas de plantas enfermas. Sin embargo, se reconoce que estos resultados se basan en un solo conjunto de datos aumentado y que los hiperparámetros de entrenamiento de la red necesitan ajustarse según el conjunto de datos.</p>
<p>Efficient crop segmentation net and novel weed detection method (Kong, X., Liu, T., Chen, X., Jin, X., Li, A., & Yu, J., 2024)</p>	<p>https://doi.org/10.1016/j.eja.2024.127367</p>	<p>Este estudio aborda los desafíos en el control preciso de malezas mediante visión por computadora, enfocado en reducir el uso de herbicidas y los costos asociados al manejo de malezas. La investigación presenta una metodología novedosa que utiliza un algoritmo de segmentación para identificar con precisión los contornos de cultivos y detectar malezas mediante técnicas de procesamiento de imágenes. Además, se desarrolla una red de segmentación basada en la arquitectura YOLO, optimizada con un nuevo backbone, un mecanismo de atención y mejoras en la técnica de fusión de características, lo que permite una mayor precisión con menores demandas computacionales.</p>

<p>SC-Net: A new strip convolutional network model for rice seedling and weed segmentation in paddy field (Liao, J., Chen, M., Zhang, K., Zhou, H., Zou, Y., Xiong, W., Zhang, S., Kuang, F., & Zhu, D., 2024)</p>	<p>https://doi.org/10.1016/j.compag.2024.108862</p>	<p>Este trabajo presenta un enfoque innovador para la segmentación semántica en la gestión automatizada de malezas en el cultivo de arroz, un factor crítico que afecta significativamente el rendimiento y la calidad del cultivo. Para superar las dificultades en la distinción entre plántulas de arroz y malezas, que suelen compartir características de textura similares, se propone SC-Net, un modelo basado en una red U-Net optimizada. Este modelo integra bloques de convolución multinivel paralelos (PMCB) y de tiras multinivel (SMCB) en su codificador y decodificador, permitiendo extraer características sobresalientes con un bajo costo computacional. Además, incorpora módulos de atención avanzada, como el módulo de fusión de características por atención (AFF) y el módulo de atención con agrupamiento por tiras (SPAM), que mejoran la percepción espacial precisa y la integración adaptativa de características.</p>
<p>An Improved UNet Lightweight Network for Semantic Segmentation of Weed Images in Corn Fields (Zuo, Y., & Li, W., 2024)</p>	<p>https://doi.org/10.32604/cmc.2024.049805</p>	<p>Este trabajo aborda los desafíos de segmentación de malezas en campos de maíz, como la similitud entre plántulas de maíz y malezas y la pérdida de detalles en los bordes de las plantas. Además, considera las limitaciones de recursos computacionales y datos en áreas remotas como tierras agrícolas. Para enfrentar estas dificultades, se propone un algoritmo mejorado de segmentación basado en U-Net. Este modelo integra una estructura residual invertida en la ruta de contracción para reducir parámetros y mejorar la extracción de características, un módulo de agrupación piramidal para mejorar la captura de información contextual y manejar problemas de pérdida de pequeños objetivos, y un mecanismo de atención tipo squeeze-and-excitation.</p>
<p>SWFormer: A scale-wise hybrid CNN-Transformer network for multi-classes weed segmentation (Jiang, H., Chen, Q., Wang, R., Du, J., & Chen, T., 2024)</p>	<p>https://doi.org/10.1016/j.jksuci.2024.102144</p>	<p>SWFormer es una red híbrida CNN-Transformer desarrollada para segmentar malezas en campos de colza, abordando las similitudes en color, forma y textura entre cultivos y malezas. La arquitectura combina la capacidad de las CNN para capturar relaciones locales con los transformers para dependencias globales. Incluye los módulos SWCC, diseñado para capturar características multiescala y ASA, que permite fusionar información de manera adaptativa y efectiva.</p>

Nota. Elaboración propia

CAPÍTULO 2

Desarrollo

La finalidad de este proyecto es desarrollar un sistema de cuantificación automática de malezas en campos de cultivo de papa mediante el uso de imágenes adquiridas por drones y la aplicación de la red neuronal convolucional U-Net. Este enfoque tiene como objetivo reducir significativamente el tiempo necesario para la cuantificación manual de las malezas, optimizando así los procesos de monitoreo y gestión en la agricultura.

2.1. Descripción del Proyecto

El sistema se compone de dos componentes principales: un servidor web, que albergará la API Flask para el preprocesamiento de datos y la inferencia con el modelo U-Net, crucial para el análisis de imágenes capturadas por el dron; y una interfaz de usuario, una aplicación móvil en Xamarin, que permitirá al usuario cargar imágenes del terreno y recibir los resultados del análisis de malezas a través de la comunicación con la API. Además, la obtención de datos se realizará mediante el dron. Todo el backend será desplegado en la nube para asegurar un uso óptimo de recursos y tiempos de respuesta eficientes de la API.

Para la construcción del sistema, se utilizó la metodología KDD (Knowledge Discovery in Databases), la cual abarca las etapas de selección, preprocesamiento, transformación y minería de datos, permitiendo así su posterior interpretación y utilización en la red neuronal convolucional U-Net. Además, para la planificación y gestión de las actividades del proyecto, se empleó la metodología Kanban, que facilita la visualización del flujo de trabajo y mejora la gestión de proyectos.

2.1.1. Planificación de actividades

La planificación de actividades prevista para desarrollo del proyecto se muestra en la tabla 4.

Tabla 4

Planificación

Actividades	Tiempo estimado
Búsqueda de terrenos de cultivos de papa	1 semana
Adecuar el terreno para la investigación sembrando las malezas correspondientes	5 semanas

Capturar las imágenes y seleccionar las más adecuadas	1 semana
Realizar la anotación manual de un conjunto representativo de imágenes usando segmentación semántica	8 semanas
Diseñar la interfaz de usuario de la aplicación móvil	1 semana
Implementar la arquitectura de la red neuronal convolucional U-Net y entrenarla con el conjunto de imágenes	1 semana
Integrar el modelo entrenado de U-Net en una API en Flask	2 semanas
Desarrollar la aplicación móvil y comunicarla con la API Flask para realizar las inferencias	2 semanas
Despliegue de la aplicación en un servidor de cloud	1 semana

Nota. Elaboración propia

2.1.2. Entregables

A continuación, se presentan los elementos entregables, que se describen en detalle en la tabla 5. Estos serán proporcionados al tutor del proyecto, el PhD. Iván García, una vez que el proyecto esté completo.

Tabla 5

Entregables

Entregable	Descripción
Modelo de la red neuronal convolucional U-Net entrenada	Modelo que presente las mejores métricas en formato h5
Data set	Conjunto de imágenes de 128px por 128px de plantas de papa y malezas etiquetadas correspondientemente
Interpretación de resultados del modelo	Análisis comparativo de las métricas del modelo entrenado

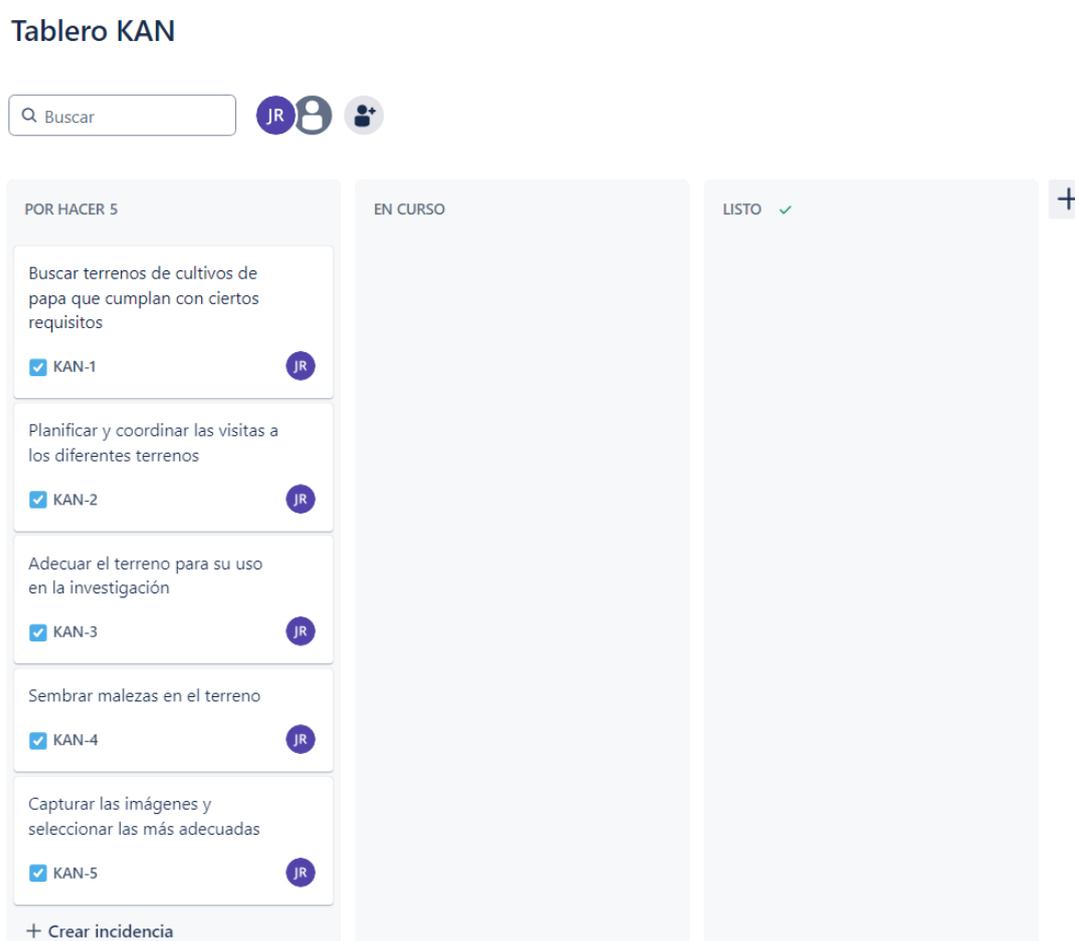
Nota. Elaboración propia

2.2. Recolección de datos

La metodología KDD inicia con la recopilación de datos, una etapa crucial en este proyecto, ya que proporciona la base sobre la cual se desarrollará y evaluará el sistema de cuantificación automática de malezas. En este caso, fue necesario iniciar con las actividades presentadas en la figura 6.

Figura 6

Actividades para la recolección de datos



Nota. Elaboración propia

Estas actividades fueron esenciales para el cumplimiento de la primera etapa de dicha metodología y son detalladas a continuación:

- **Búsqueda de terrenos de cultivo de papa:** Se llevó a cabo una búsqueda de terrenos de cultivos de papa que cumplieran con ciertos requisitos específicos, tales

como la accesibilidad y la presencia mínima de las malezas previamente seleccionadas. Cabe recalcar que en un inicio se planeaba ir a la granja “La Pradera” de la Universidad Técnica del Norte, pero esto no pudo realizarse porque el administrador de esta nos informó que no estaba disponible ningún terreno con las especificaciones que necesitábamos.

- **Planificación y coordinación de visitas:** Una vez seleccionados los terrenos, se planificaron y coordinaron las visitas a cada uno de ellos. Esto incluyó el establecimiento de fechas específicas con el grupo de tesis, tutor y asesor del proyecto. Los terrenos visitados se detallan en la tabla 6.

Tabla 6

Terrenos visitados

Terreno	Ubicación	Fecha	Coordenada GPS	Propietario
		20/05/2023		
1	San Gabriel, comunidad de Chutan Bajo	28/05/2023 04/06/2023	77°51'03"W	Propiedad del padre del Ing. Marco Pusdá
2	San Gabriel	04/06/2023	77°48'403"W	Propiedad anónima
3	Cuesaca, Avenida de la Prehistoria	17/06/2023	77°52'40"W	Propiedad del tío del estudiante Kevin Vinueza
4	El Ángel, barrio San Vicente	16/12/2023	0°38'21"N	Propiedad anónima
5	El Ángel, barrio San Vicente	16/12/2023	77°55'59"W	Propiedad anónima
6	El Ángel, barrio San Vicente	16/12/2023	77°55'59"W	Propiedad anónima
7	El Ángel, barrio San Vicente	16/12/2023	0°38'13"N	Propiedad anónima
8	Cotacachi, comunidad Tunibamba de Bellavista	11/05/2024	0°19'00.1"N	Propiedad de los padres del estudiante Luis Perugachi
		17/05/2024	78°15'56.2"W	

Nota. Elaboración propia

Es importante destacar que los terrenos 1, 2 y 3 fueron visitados por el grupo de tesis anterior para llevar a cabo investigaciones relacionadas.

- **Adecuación del terreno:** Para preparar el terreno 8 ubicado en Cotacachi, se llevaron a cabo diversas actividades de adecuación. Esto incluyó la limpieza de la superficie, la delimitación de áreas específicas para la siembra y la preparación del suelo con maquinaria de arado. En la figura 7 se puede visualizar el terreno preparado para sembrar.

Figura 7

Terreno n°8 adecuado



Nota. Elaboración propia

- **Siembra de malezas:** En el terreno anteriormente adecuado, se procedió a la siembra de malezas. Este paso fue esencial para crear un entorno controlado en el que se pudiera evaluar la eficacia del sistema de cuantificación de malezas. Fue necesario realizar esta actividad para equilibrar el número de imágenes por cada clase en el conjunto de datos. En total, se consideran tres tipos de malezas: lengua de vaca, diente de león y kikuyo, además de las plantas de papa. Por esta razón, fue indispensable plantar las dos últimas, ya que se requerían más imágenes de estas malezas. En la figura 8 se evidencia este proceso.

Figura 8

Siembra de malezas en el terreno n°8



Nota. Elaboración propia

- **Capturar imágenes:** Las imágenes fueron recopiladas con ayuda del dron DJI Mavic 2 Pro, este fue configurado en la aplicación DroneDeploy 5.7, siendo esta la más compatible con el equipo y brindando funciones útiles a la hora de la captura de imágenes. Las especificaciones de vuelo fueron las siguientes:
 - **Altura de vuelo:** 9 metros
 - **Velocidad de vuelo:** 1 m/s
 - **Modo de captura:** Video
 - **Resolución de imagen:** 5472px por 3648px

Estas configuraciones permitieron la captura de videos desde los cuales se extrajeron los frames correspondientes para el análisis. En la figura 9 se puede apreciar el aterrizaje del dron una vez recorrido el terreno en Cotacachi.

Figura 9

Aterrizaje del dron DJI Mavic 2 Pro



Nota. Elaboración propia

En la tabla 7 se detalla el número de imágenes obtenidas en cada uno de los terrenos y la etapa de crecimiento correspondiente.

Tabla 7

Imágenes obtenidas por cada terreno

Terreno	Ubicación	Fecha	Numero de imágenes	Etapas de crecimiento
1	San Gabriel, comunidad	20/05/2023	539	20 días aprox.
	de Chutan Bajo,	28/05/2023	409	28 días aprox.
	77°51'03"W	04/06/2023	926	35 días aprox.
2	San Gabriel, 77°48'403"W	04/06/2023	197	28 días aprox.
3	Cuesaca, Avenida de la Prehistoria, 77°52'40"W	17/06/2023	209	30 días aprox.
4	El Ángel, barrio San	16/12/2023	61	30 días aprox.

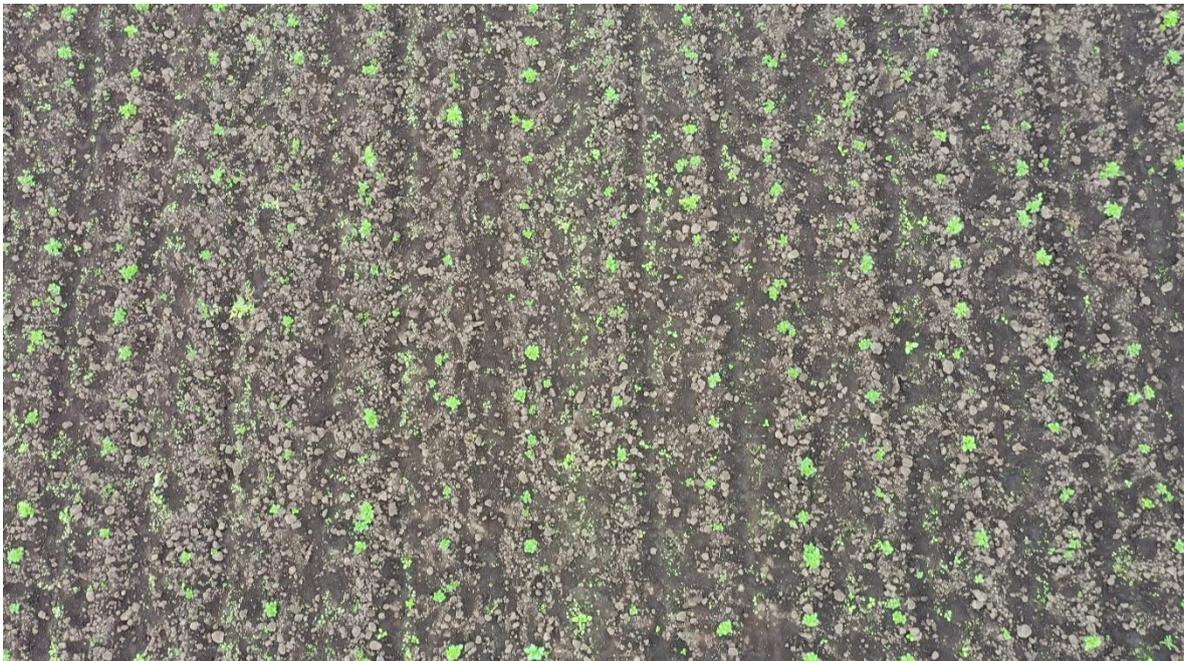
	Vicente, 0o38'21"N			
5	El Ángel, barrio San Vicente, 77°55'59"W	16/12/2023	28	20 días aprox.
6	El Ángel, barrio San Vicente, 77°55'59"W	16/12/2023	53	30 días aprox.
7	El Ángel, barrio San Vicente, 0°38'13"N	16/12/2023	25	15 días aprox.
				15 - 20 días aprox.
8	Cotacachi, comunidad Tunibamba de Bellavista, 0°19'00.1"N 78°15'56.2"W	11/05/2024 17/05/2024	603 760	20 - 25 días aprox.
TOTAL			3810	

Nota. Elaboración propia

En las figuras 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 y 20 se muestra una imagen capturada por cada terreno, en el caso del terreno 1 y 8 se muestra por cada fecha visitada.

Figura 10

Imagen capturada en el terreno 1 en la primera visita el 20/05/2023



Nota. Elaboración propia

Figura 11

Imagen capturada en el terreno 1 en la segunda visita el 28/05/2023



Nota. Elaboración propia

Figura 12

Imagen capturada en el terreno 1 en la tercera visita el 04/06/2023



Nota. Elaboración propia

Figura 13

Imagen capturada en el terreno 2 el 04/06/2023



Nota. Elaboración propia

Figura 14

Imagen capturada en el terreno 3 el 04/06/2023



Nota. Elaboración propia

Figura 15

Imagen capturada en el terreno 4 el 16/12/2023



Nota. Elaboración propia

Figura 16

Imagen capturada en el terreno 5 el 16/12/2023



Nota. Elaboración propia

Figura 17

Imagen capturada en el terreno 6 el 16/12/2023



Nota. Elaboración propia

Figura 18

Imagen capturada en el terreno 7 el 16/12/2023



Nota. Elaboración propia

Figura 19

Imagen capturada en el terreno 8 en la primera visita el 11/05/2024



Nota. Elaboración propia

Figura 20

Imagen capturada en el terreno 8 en la segunda visita el 17/05/2024



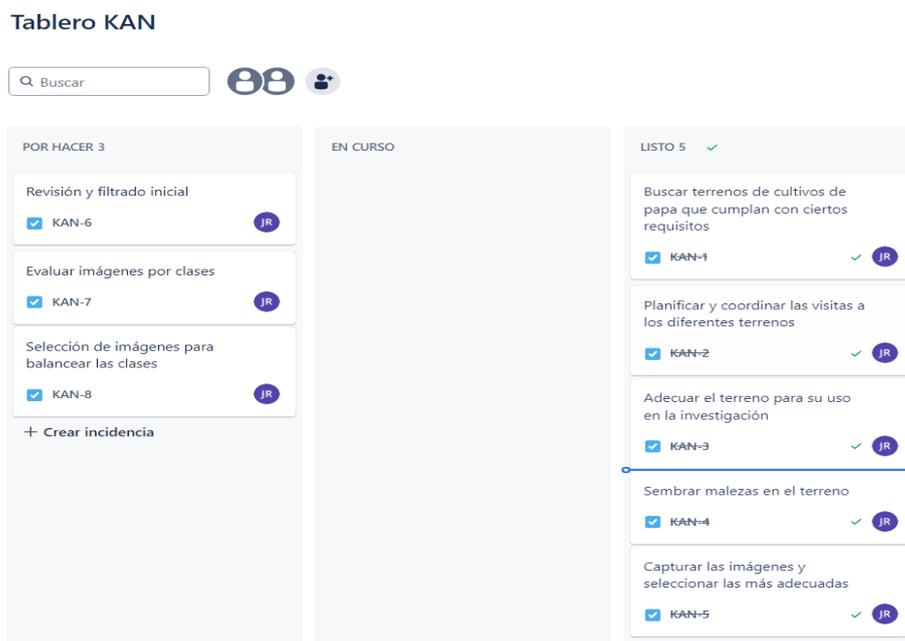
Nota. Elaboración propia

2.3. Selección de imágenes

La siguiente etapa en la metodología KDD es la selección de datos, una etapa importante en la construcción del conjunto de datos para el entrenamiento del modelo U-Net, garantizando la calidad de la información que será utilizada posteriormente. Se planificó diversas actividades las cuales se visualizan en la figura 21.

Figura 21

Actividades para la etapa de selección de imágenes



Nota. Elaboración propia

La primera actividad es la revisión y filtrado inicial de las imágenes capturadas, implicó evaluar cada imagen en términos de nitidez y claridad, con un enfoque particular en la visibilidad de las malezas. Las imágenes que no cumplían con los criterios de calidad fueron descartadas. Esta actividad fue fundamental para asegurar que el conjunto de datos final contenga imágenes de alta calidad que faciliten la detección precisa de las malezas.

Las imágenes que fueron descartadas fueron generalmente los primeros frames y en imágenes las cuales abarquen las mismas plantas y malezas, además de descartar los límites del terreno. En la figura 22 se puede observar ejemplos de imágenes descartadas aplicando los criterios anteriormente mencionados.

Figura 22

Imágenes que fueron descartadas en la fase de selección



Nota. Elaboración propia

La siguiente actividad consistió en revisar las clases presentes en el conjunto de datos y evaluar la necesidad de balancearlas. Dado que el dataset se compartía con el tesista Kevin Vinueza que ya había culminado su trabajo (2024) , fue necesario identificar las clases de malezas que tenían un menor número de imágenes. En la tabla 8 se muestran las clases y el número de imágenes por cada una.

Tabla 8

Imágenes por cada clase

Clase	Número de imágenes
Papa	500
Lengua de vaca	500
Diente de león	160
Kikuyo	300
Otras	40

Nota. Extraído de: (Vinueza K., 2024).

Esta evaluación ayudó a determinar que clases necesitaban más imágenes para lograr un conjunto de datos equilibrado. Según la tabla sabemos que las clases “diente de león”, “kikuyo” y “otras” son las clases que necesitan ser balanceadas dentro del dataset y de las cuales se va a trabajar en las siguientes actividades.

Finalmente, se seleccionaron las imágenes en las cuales había presencia de las malezas anteriormente mencionadas. Este proceso implicó seleccionar el número adecuado de imágenes para crear subimágenes correspondientes a cada clase a balancear. En la tabla 9 se evidencia este proceso.

Tabla 9

Imágenes seleccionadas para balancear las clases

Clase	Número de imágenes en el dataset	Número de imágenes seleccionadas	Total
Papa	500	0	500
Lengua de vaca	500	0	500
Diente de león	160	340	500
Kikuyo	300	200	500
Otras	40	60	100
TOTAL	1500	600	2100

Nota. Elaboración propia

2.4. Preprocesamiento y transformación de datos

La siguiente fase es el preprocesamiento y la transformación de datos con el fin de preparar las imágenes que se utilizarán en el entrenamiento del modelo U-Net. Este proceso incluyó la extracción de subimágenes y el ajuste de sus dimensiones para asegurar que se adapten a los requisitos del modelo.

Inicialmente, se seleccionaron imágenes completas de 5472 x 3648 píxeles durante la fase anterior. A partir de estas imágenes, se extrajeron subimágenes de 250 x 250 píxeles. Este tamaño fue elegido porque es ideal para abarcar una planta completa, lo cual es esencial para la tarea de segmentación precisa que realiza el modelo U-Net.

Este enfoque está alineado con el trabajo realizado por el tesista Kevin Vinueza. En su trabajo de investigación, se utilizó la arquitectura ResNext50, que requiere imágenes de entrada de dimensiones pequeñas, específicamente 128 x 128 píxeles. Ambos trabajos coinciden en la necesidad de extraer pequeñas subimágenes para mejorar el rendimiento del

modelo sin perder información de las imágenes originales. Según el estudio realizado por T et al. (2019), las subimágenes de 250 x 250 píxeles son adecuadas para cubrir toda el área de una planta de papa o maleza, lo cual justifica su elección en este proyecto.

El proceso de extracción de subimágenes se realizó utilizando el software Adobe Photoshop, aprovechando las herramientas disponibles para realizar esta tarea de manera eficiente y precisa. La herramienta de rectángulo en Photoshop fue configurada para crear cuadrados de 250 x 250 píxeles, que se posicionaron manualmente sobre las plantas para cubrir la mayor parte de cada una de ellas. Sin embargo, para esta tarea se puede optar por otros software más accesibles y gratuitos como lo es Canva, que, a pesar de contar con su versión de paga, la versión gratuita es suficiente para realizar esta actividad.

En la figura 23 se muestra como fue este proceso en Adobe Photoshop.

Figura 23

Extracción de subimágenes de 250 x 250 píxeles



Nota. Elaboración propia

Una vez culminada esta tarea se redimensionó cada imagen a 128 x 128 píxeles en el mismo software Adobe Photoshop para su posterior uso en el modelo U-Net. Al no ser una redimensión tan significativa se pudo conservar una cantidad considerable de información de cada imagen, puesto que la reducción de tamaño implica que los píxeles se irán reduciendo y por ende perdiendo información que podría ser importante al entrenar nuestro modelo.

En la tabla 10 se observa un resumen de la fase de preprocesamiento de las imágenes.

Tabla 10

Resumen del preprocesamiento de imágenes

Etapa	Descripción
Selección de imágenes	Imágenes seleccionadas en la fase anterior
Extracción de subimágenes	Subimágenes de 250 x 250 píxeles extraídas de imágenes de 5472 x 3648 píxeles
Redimensión de subimágenes	Las subimágenes extraídas fueron redimensionadas a 128 x 128 píxeles
Herramienta utilizada	Adobe Photoshop
Técnica de extracción	Manual, utilizando la herramienta de rectángulo

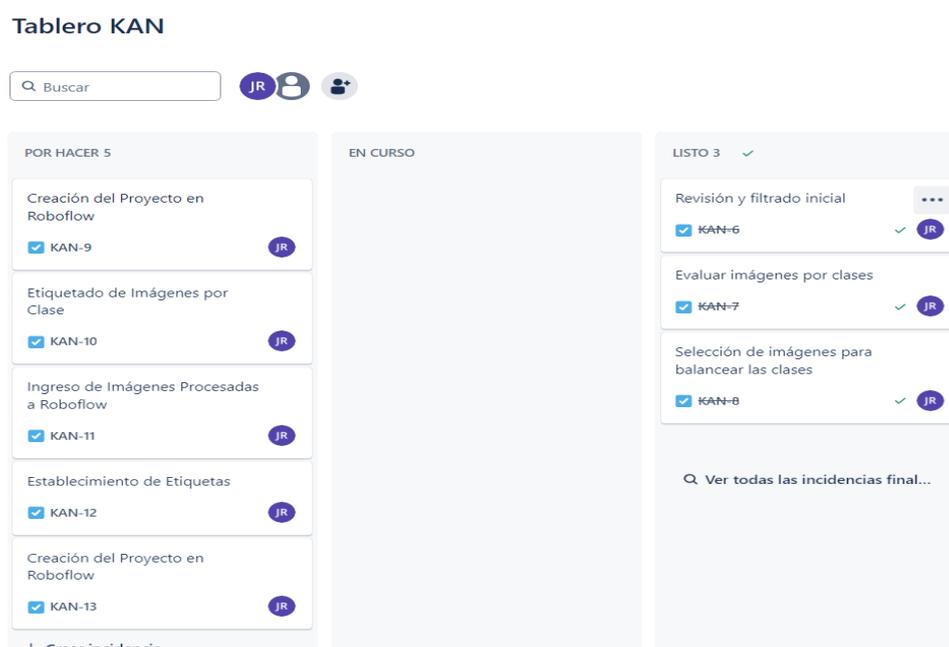
Nota. Elaboración propia

2.5. Anotación manual

La siguiente fase en la metodología KDD es la anotación de los datos, esta fase es un paso importante en la preparación del conjunto de datos, ya que permite etiquetar las imágenes con precisión para el entrenamiento del modelo de U-Net usando la segmentación semántica. En este proyecto, la herramienta Roboflow se utilizó para llevar a cabo este proceso. Las actividades realizadas durante esta fase se describen en la figura 24.

Figura 24

Actividades para la anotación manual en Roboflow



Nota. Elaboración propia

El proceso comenzó con la creación de un proyecto en Roboflow específicamente para la tarea de segmentación semántica. Esta plataforma proporciona herramientas avanzadas para la anotación y gestión de imágenes, facilitando la organización y etiquetado de grandes volúmenes de datos.

A continuación, se ingresaron las imágenes previamente revisadas y procesadas a la plataforma. Estas imágenes fueron seleccionadas y preparadas en etapas anteriores, asegurando que estuvieran listas para el proceso de anotación. Para proceder con la anotación, se establecieron las etiquetas descritas en la tabla 11.

Tabla 11

Clases definidas por cada tipo de maleza y cultivo

Índice	Clase	Descripción
0	Background	Fondo de la imagen
1	Cow-Tongue	Planta de lengua de vaca
2	Dandelion	Planta de diente de león
3	Kikuyo	Planta de kikuyo
4	Other	Otras plantas existentes en las imágenes
5	Potato	Planta de papa

Nota. Elaboración propia

Estas etiquetas fueron seleccionadas para abarcar las principales clases de interés en el estudio y asegurar una correcta segmentación de las imágenes.

La última actividad en el proceso de anotación fue el etiquetado de cada imagen según las clases definidas. Este paso se realizó de manera manual utilizando las herramientas de segmentación semántica proporcionadas por Roboflow. Durante esta actividad, se dibujaron máscaras precisas alrededor de cada objeto en las imágenes, asignando la etiqueta correspondiente a cada región.

En la figura 25 se muestra un ejemplo de una imagen anotada en Roboflow y en la figura 26 se puede observar una imagen con su respectiva máscara generada.

Figura 25

Imagen anotada en Roboflow



Nota. Elaboración propia

Figura 26

Imagen con su máscara de segmentación semántica generada



Nota. Elaboración propia

Una vez completada la anotación de todas las imágenes, se generó el conjunto de datos anotado con sus respectivas máscaras de segmentación semántica, además de separarlas en un conjunto de entrenamiento y un conjunto de validación con un volumen del 80% y 20% respectivamente con respecto al número total de imágenes. En la tabla 12 se

detalla el número total de imágenes por clase para estos dos conjuntos de datos. Una opción de Roboflow permite contabilizar el número de individuos por cada categoría en el dataset, esta información se describe en la tabla 13 en comparación con los individuos del dataset original.

Tabla 12

Número de imágenes por clase para el conjunto de entrenamiento y validación

Tipo de planta	Conjunto de entrenamiento (80%)	Conjunto de validación (20%)
Lengua de vaca	400	100
Diente de león	400	100
Kikuyo	400	100
Papa	400	100
Otras	80	20
TOTAL	1680	420

Nota. Elaboración propia

Tabla 13

Número de individuos por cada clase en comparación con el dataset original (desbalanceado)

Tipo de planta	Dataset Original		Dataset Nuevo	
	Cantidad de individuos	Porcentaje de representación con respecto a los individuos totales	Cantidad de individuos	Porcentaje de representación con respecto a los individuos totales
Lengua de vaca	978	14.43%	978	12.05%
Diente de león	321	4.73%	883	10.87%
Kikuyo	2670	39.41%	3060	37.68%
Papa	1808	26.69%	1808	22.27%
Otras	997	14.71%	1390	17.12%
TOTAL	6774	100%	8119	100%

Nota. Elaboración propia

2.6. Aumentación de datos

Para mejorar el rendimiento y la robustez del modelo de segmentación semántica U-Net, se implementaron varias técnicas de aumentación de datos utilizando Opencv y Python, con el objetivo de incrementar el número total de imágenes por cada clase y diversificar el conjunto de datos de entrenamiento. Las técnicas aplicadas y el número de imágenes obtenidas se detallan en la tabla 14.

Tabla 14

Técnicas de Data Aumentation y número de imágenes obtenidas

Técnica de Data Aumentation	Descripción	Número de imágenes obtenidas
Aumento del brillo 25%	Incremento del brillo de la imagen en un 25%	1680
Disminución del brillo 25%	Reducción del brillo de la imagen en un 25%	1680
Desenfoque gaussiano (5x5)	Aplicación de un filtro de desenfoque gaussiano con kernel 5x5	1680
Rotación 90° + Aumento de brillo 25%	Rotación de 90 grados en sentido horario con aumento del brillo en un 25%	1680
Rotación 180° + Disminución de brillo 25%	Rotación de 180 grados en sentido horario con disminución del brillo en un 25%	1680
Rotación 270°	Rotación de 270 grados en sentido horario	1680
Imágenes Originales	Imágenes sin ninguna transformación adicional	1680
TOTAL		11 760

Nota. Elaboración propia

Estas técnicas de aumentación no solo incrementan el tamaño del conjunto de datos, sino que también introducen variabilidad, permitiendo al modelo aprender a reconocer las malezas en condiciones diversas, mejorando así su capacidad de generalización y robustez

en escenarios reales. La implementación de estas técnicas de aumentación de datos es esencial para fortalecer el rendimiento del modelo U-Net en la segmentación semántica de malezas en imágenes agrícolas. En la Tabla 15 se presenta el número de individuos por cada clase después de aplicar las técnicas de data augmentation. Estas técnicas permitieron aumentar el número de individuos en las clases con menor representación, logrando así un dataset final balanceado.

Tabla 15

Número de individuos por cada clase en el dataset balanceado

Tipo de planta	Dataset Balanceado	
	Cantidad de individuos	Porcentaje de representación con respecto a los individuos totales
Lengua de vaca	4234	16.44%
Diente de león	4595	17.84%
Kikuyo	5829	22.63%
Papa	5829	22.63%
Otras	5270	20.46%
TOTAL	25757	100%

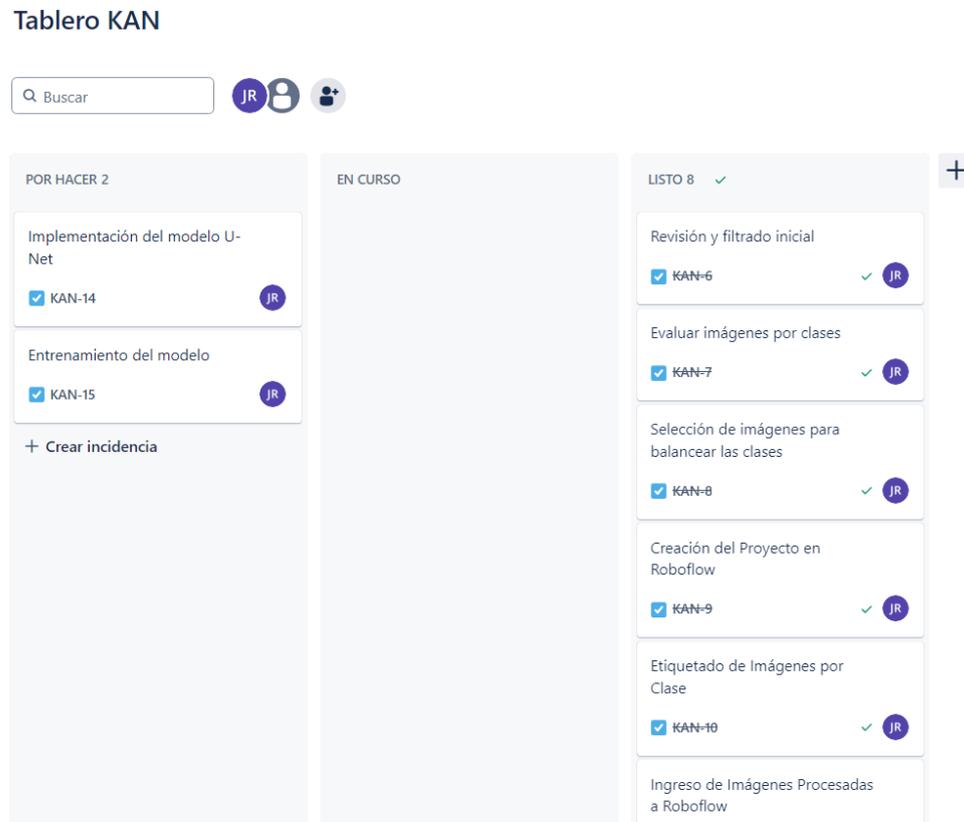
Nota. Elaboración propia

2.7. Minería de datos

En esta fase, se procedió a la construcción y entrenamiento de la red neuronal convolucional utilizando la arquitectura U-Net. La arquitectura del modelo U-Net fue seleccionada para la tarea de segmentación semántica debido a su eficacia comprobada en la segmentación de imágenes médicas y su capacidad para capturar tanto características de bajo nivel como de alto nivel. En la figura 27 se detalla las actividades para llevar a cabo esta fase.

Figura 27

Actividades para la minería de datos



Nota. Elaboración propia

2.7.1. Implementación del algoritmo U-Net

El modelo U-Net se implementó utilizando TensorFlow y Keras. La red consta de dos partes principales: el encoder y el decoder.

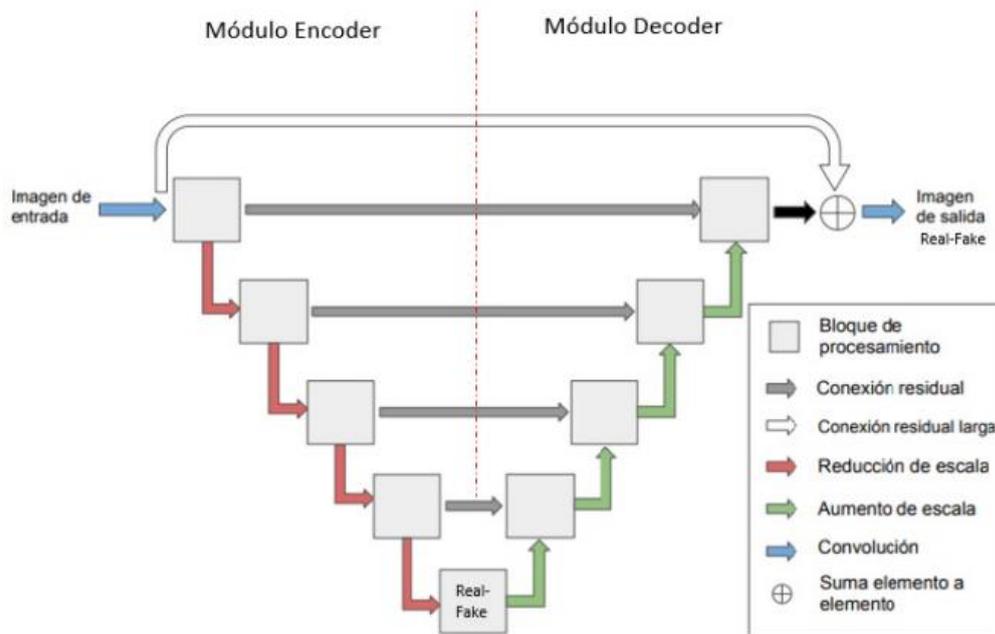
- 1. Encoder (Contracción):** Esta parte reduce la dimensionalidad de la imagen, extrayendo características relevantes a través de una serie de capas convolucionales y de pooling. El encoder está compuesto por:
 - a. Varias capas de convolución con activación ReLU y padding 'same'.
 - b. Capas de max pooling para reducir la dimensionalidad espacial.
 - c. Dropout para prevenir el sobreajuste en las capas más profundas.
- 2. Decoder (Expansión):** Esta parte aumenta la dimensionalidad de la imagen a su tamaño original, combinando características de baja y alta resolución. El decoder incluye:
 - a. Capas de convolución transpuesta para aumentar la dimensionalidad.
 - b. Operaciones de concatenación para combinar características de la misma escala del encoder y el decoder.

c. Capas de convolución para refinar la imagen segmentada.

El modelo finaliza con una capa de convolución con 6 filtros y activación softmax para obtener la segmentación en 6 clases diferentes. En la figura 28 se muestra la arquitectura implementada.

Figura 28

Arquitectura U-Net



Nota. Modificado de: (Ronneberger et al., 2015)

La tabla 16 muestra las capas del modelo U-Net, sus tipos, tamaños de salida y el número de parámetros. La red tiene una estructura simétrica con una fase de contracción seguida de una fase de expansión, lo que permite una segmentación precisa al combinar características de diferentes niveles.

Tabla 16

Resumen de la arquitectura del modelo U-Net

Capa	Tipo	Tamaño de Salida
Input	Entrada	(128, 128, 3)
Conv2D	Convolución	(128, 128, 64)
Conv2D	Convolución	(128, 128, 64)

MaxPooling2D	Max Pooling	(64, 64, 64)
Conv2D	Convolución	(64, 64, 128)
Conv2D	Convolución	(64, 64, 128)
MaxPooling2D	Max Pooling	(32, 32, 128)
Conv2D	Convolución	(32, 32, 256)
Conv2D	Convolución	(32, 32, 256)
MaxPooling2D	Max Pooling	(16, 16, 256)
Conv2D	Convolución	(16, 16, 512)
Conv2D	Convolución	(16, 16, 512)
Dropout	Dropout	(16, 16, 512)
MaxPooling2D	Max Pooling	(8, 8, 512)
Conv2D	Convolución	(8, 8, 1024)
Conv2D	Convolución	(8, 8, 1024)
Dropout	Dropout	(8, 8, 1024)
Conv2DTranspose	Convolución Transp.	(16, 16, 512)
Conv2D	Convolución	(16, 16, 512)
Concatenate	Concatenación	(16, 16, 1024)
Conv2D	Convolución	(16, 16, 512)
Conv2D	Convolución	(16, 16, 512)
Conv2DTranspose	Convolución Transp.	(32, 32, 256)
Conv2D	Convolución	(32, 32, 256)
Concatenate	Concatenación	(32, 32, 512)
Conv2D	Convolución	(32, 32, 256)
Conv2D	Convolución	(32, 32, 256)
Conv2DTranspose	Convolución Transp.	(64, 64, 128)
Conv2D	Convolución	(64, 64, 128)
Concatenate	Concatenación	(64, 64, 256)
Conv2D	Convolución	(64, 64, 128)
Conv2D	Convolución	(64, 64, 128)
Conv2DTranspose	Convolución Transp.	(128, 128, 64)
Conv2D	Convolución	(128, 128, 64)
Concatenate	Concatenación	(128, 128, 128)
Conv2D	Convolución	(128, 128, 64)
Conv2D	Convolución	(128, 128, 64)
Conv2D	Convolución	(128, 128, 6)

2.7.2. Entrenamiento del modelo

El entrenamiento del modelo se realizó en Google Colab, para lo cual fue necesario adquirir la versión PRO, que ofreció un mayor poder de cómputo en comparación con la versión gratuita. Las especificaciones detalladas de las GPUs utilizadas están disponibles en la Tabla 17. Google Colab es una herramienta gratuita de Google que permite a los usuarios escribir y ejecutar código Python en el navegador, y ofrece ventajas como acceso a GPUs y TPUs, un entorno basado en la nube que elimina la necesidad de instalar software adicional, colaboración en tiempo real, integración con Google Drive, preconfiguración con bibliotecas comunes, soporte para Markdown, ejecución de cualquier código Python, instalación fácil de paquetes adicionales y una interfaz intuitiva con buena documentación, lo que la hace ideal para proyectos de aprendizaje automático y análisis de datos (Google, 2024).

El conjunto de datos utilizado para el entrenamiento y la validación se cargó desde Google Drive. Este conjunto consta de imágenes y sus correspondientes máscaras, necesarias para la tarea de segmentación. Las imágenes fueron preprocesadas y convertidas a matrices NumPy para ser utilizadas en el modelo. A continuación, en la figura 29 se visualiza una muestra de las imágenes y sus máscaras para verificar la correcta carga y preprocesamiento.

Tabla 17

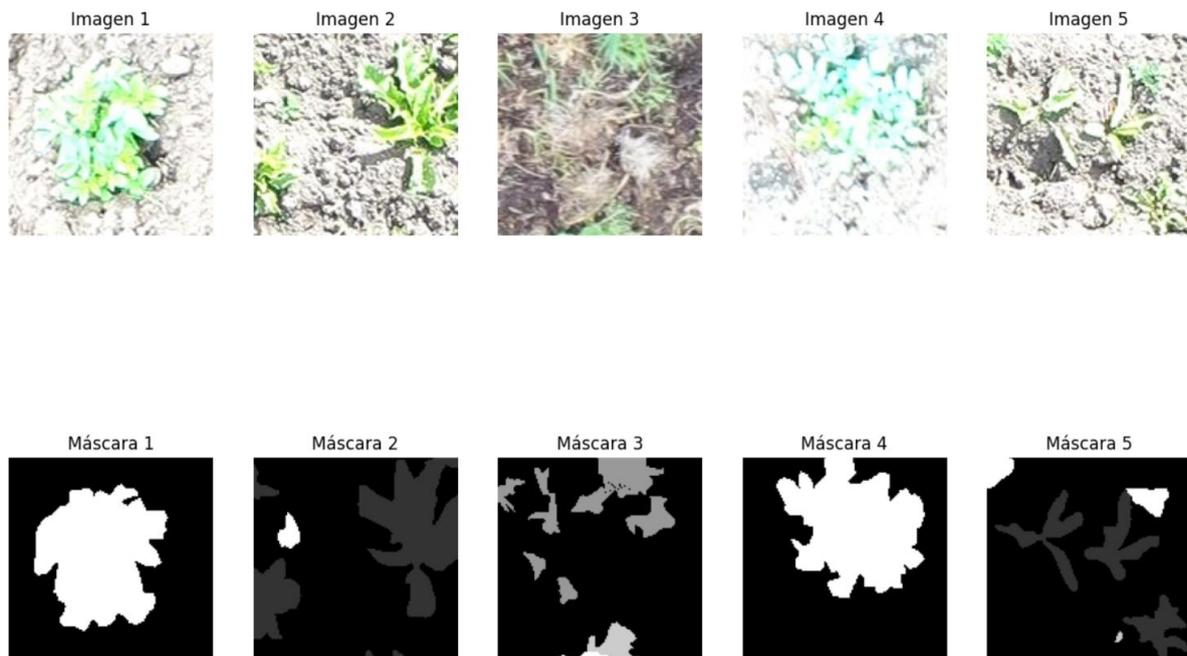
Especificaciones de las GPUs en Google Colab

Versión	Tipo de GPU	RAM del sistema	RAM GPU	Disco
Gratuita	Nvidia Tesla T4	12.7 GB	15 GB	201.2 GB
Pro (L4)	Nvidia L4	53 GB	22.5 GB	201.2 GB
Pro (A100)	Nvidia A100	83.5 GB	40 GB	201.2 GB

Nota. Google, 2024

Figura 29

Visualización de datos de entrenamiento



Nota. Elaboración propia

El modelo se compiló utilizando el optimizador Adam y la función de pérdida de entropía cruzada categórica dispersa, adecuada para la tarea de segmentación multicategoría. Se implementó el callback de EarlyStopping para detener el entrenamiento cuando la pérdida de validación no mejorara después de 5 épocas consecutivas, restaurando los mejores pesos del modelo y el callback ReduceLROnPlateau para reducir la tasa de aprendizaje cuando la precisión de validación no mejore por un cierto número de épocas.

El modelo fue entrenado durante un máximo de 100 épocas con un tamaño de lote de 32. Los datos de validación se utilizaron para evaluar el rendimiento del modelo durante el entrenamiento.

2.7.3. Resultados del entrenamiento

Los resultados del entrenamiento se almacenaron y se visualizan a través de gráficos que muestran la precisión y la pérdida tanto en el conjunto de datos de entrenamiento como en el de validación a lo largo de las épocas. Dichos resultados se presentan en el tercer capítulo.

Además, en la tabla 18 se detalla los resultados del entrenamiento en cuanto a sus parámetros.

Tabla 18*Parámetros del modelo entrenado*

Tipo	Filtros	Tamaño del Kernel	Activación	Parámetros
Convolución	64	3x3	ReLU	1,792
Convolución	64	3x3	ReLU	36,928
MaxPooling2D	-	2x2	-	0
Convolución	128	3x3	ReLU	73,856
Convolución	128	3x3	ReLU	147,584
MaxPooling2D	-	2x2	-	0
Convolución	256	3x3	ReLU	295,168
Convolución	256	3x3	ReLU	590,080
MaxPooling2D	-	2x2	-	0
Convolución	512	3x3	ReLU	1,180,160
Convolución	512	3x3	ReLU	2,359,808
Dropout	-	-	-	0
MaxPooling2D	-	2x2	-	0
Convolución	1024	3x3	ReLU	4,719,616
Convolución	1024	3x3	ReLU	9,438,208
Dropout	-	-	-	0
Conv2DTranspose	512	2x2	ReLU	2,097,664
Convolución	512	3x3	ReLU	1,049,088
Concatenate	-	-	-	0
Convolución	512	3x3	ReLU	4,719,104
Convolución	512	3x3	ReLU	2,359,808
Conv2DTranspose	256	2x2	ReLU	524,544
Convolución	256	3x3	ReLU	262,400
Concatenate	-	-	-	0
Convolución	256	3x3	ReLU	1,179,904
Convolución	256	3x3	ReLU	590,080
Conv2DTranspose	128	2x2	ReLU	131,200
Convolución	128	3x3	ReLU	65,664
Concatenate	-	-	-	0
Convolución	128	3x3	ReLU	295,040
Convolución	128	3x3	ReLU	147,584
Conv2DTranspose	64	2x2	ReLU	32,832
Convolución	64	3x3	ReLU	16,448

Concatenate	-	-	-	0
Convolución	64	3x3	ReLU	73,792
Convolución	64	3x3	ReLU	36,928
Convolución	6	3x3	Softmax	3,462
Total de parámetros:	97,286,228 (371.12 MB)			
Parámetros entrenados:	32,428,742 (123.71 MB)			
Parámetros no entrenados:	0 (0.00 B)			
Parámetros del optimizador:	64,857,486 (247.41 MB)			

Nota. Elaboración propia

Finalmente, el modelo entrenado se guardó en Google Drive para su uso futuro. Esta implementación del modelo U-Net proporciona una solución eficiente para la segmentación de imágenes, con aplicaciones potenciales en diversos campos que requieren análisis de imágenes.

2.7.4. Variaciones del modelo

En el presente trabajo, se han explorado y desarrollado variaciones del modelo original de U-Net para mejorar la precisión y eficiencia en la segmentación de malezas en imágenes de cultivos agrícolas. Estas variaciones se basan en enfoques recientes en el campo de la visión por computadora, tal como se describe en el estudio titulado “Evaluación de modelos convolucionales en la segmentación de vasos sanguíneos cerebrales” de Oliver Jonathan Quintana-Quintana, Marco Antonio Aceves-Fernández, Iván Villanueva-Naquid y Saúl Tovar-Arriaga. El estudio mencionado propone diferentes arquitecturas para mejorar la segmentación en imágenes médicas, lo cual inspiró la adaptación de dichas técnicas a la segmentación de malezas.

Una de las innovaciones de este trabajo es la implementación de una U-Net con un módulo de atención y bloques residuales (Resblocks). Esta variante busca optimizar la capacidad del modelo para enfocarse en características relevantes en las imágenes, mejorando así la precisión en la segmentación. La atención permite al modelo resaltar regiones importantes de la imagen, mientras que los Resblocks ayudan a mitigar el problema de la degradación en redes más profundas, facilitando el aprendizaje y la transferencia de características. Adicionalmente, se ha incorporado una versión modificada del modelo U-Net basada en el trabajo de Shanwen Zhang y Chuanlei Zhang, titulado “Modified U-Net for plant diseased leaf image segmentation”. Esta arquitectura busca mejorar aún más la capacidad de segmentación específicamente para imágenes de hojas de plantas afectadas por enfermedades, lo que añade un enfoque adicional a la mejora de la precisión en este trabajo

de investigación.

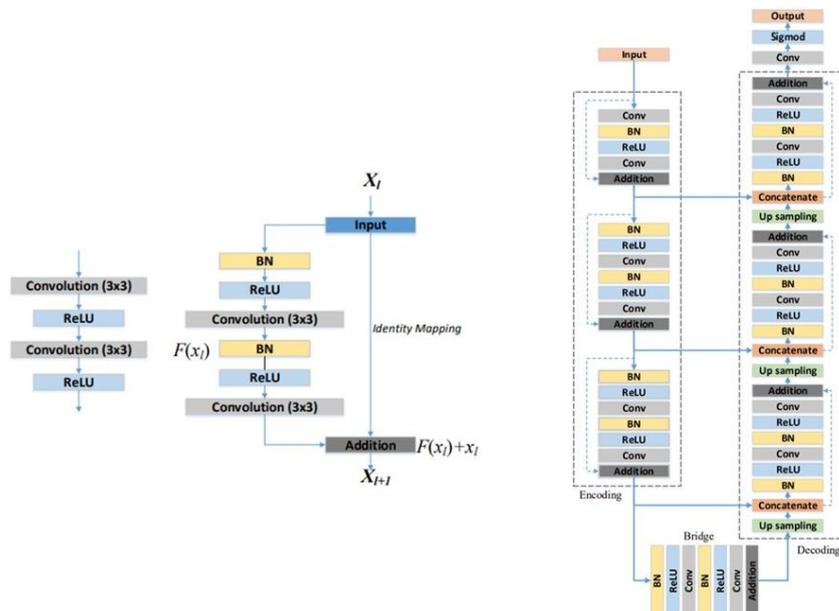
Las variaciones del modelo U-Net desarrolladas incluyen la U-Net Residual, la U-Net Doble, U-NetAtt y MU-Net detalladas en los siguientes apartados.

U-Net Residual

La arquitectura U-Net Residual se propone como una mejora de la U-Net clásica mediante la implementación de bloques residuales. Los bloques residuales facilitan el entrenamiento profundo al mitigar los problemas de degradación del gradiente. En un bloque residual, la entrada se suma a la salida de una o más capas intermedias, lo que permite que la red aprenda de manera más efectiva al mantener la integridad de la información a través de múltiples capas. La estructura del U-Net Residual mantiene la arquitectura de codificación y decodificación de la U-Net original, pero introduce conexiones residuales en varias capas del modelo. Estas conexiones permiten que la información se propague directamente desde una capa a otra sin ser completamente transformada, lo que facilita la capacitación de la red y mejora la precisión de las predicciones. En la figura 30 se muestra la arquitectura que maneja.

Figura 30

Arquitectura de U-Net Residual



Nota. Extraído de: (Quintana, Aceves, Villanueva, Tovar, 2021)

Los parámetros usados para entrenar el modelo de U-Net Residual se detallan en la tabla 19.

Tabla 19*Parámetros de entrenamiento de U-Net Residual*

Parámetro	Valor
Tamaño de entrada	128 x 128 x 3
Optimizador	Adam (learning rate = 0.001)
Función de pérdida	Sparse Categorical Crossentropy
Tamaño de lote	16
Épocas	100
Callback	EarlyStopping ReduceLROnPlateau

Nota. Elaboración propia

El total de parámetros entrenados del modelo fueron de 134,901,524 lo que supone un tamaño de 514.61 MB, siendo 44,967,174 (171.54 MB) los parámetros entrenados y 89,934,350 (343.07 MB) los parámetros del optimizador.

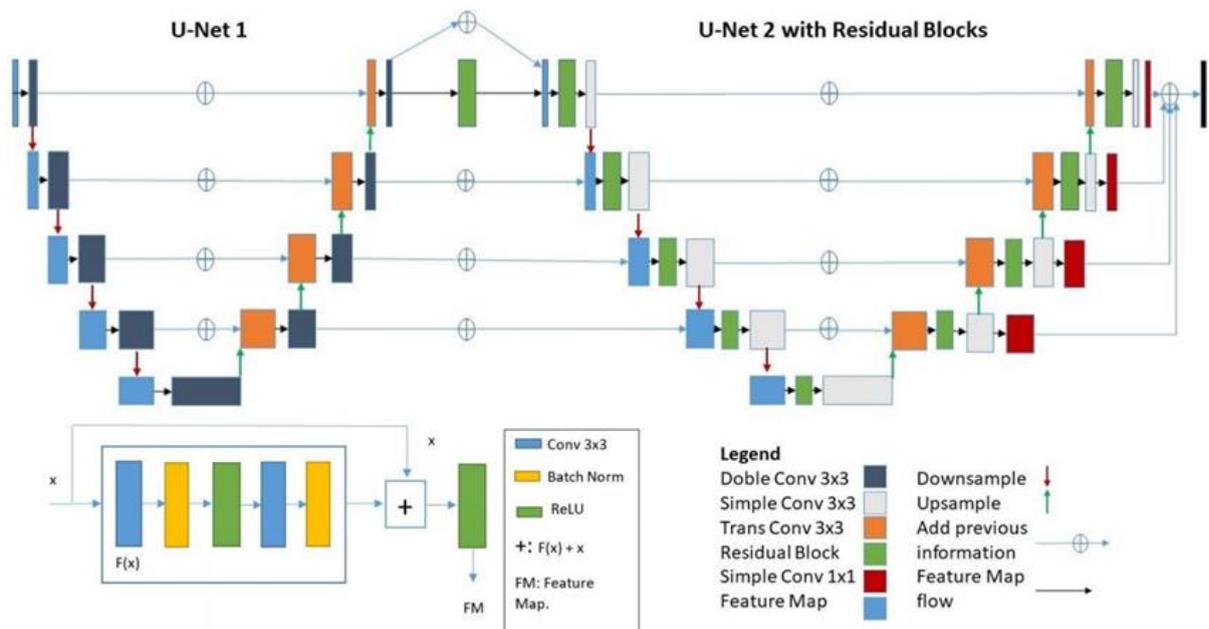
U-Net Doble

La U-Net Doble es una variación que combina una U-Net convencional con una U-Net Residual en una configuración híbrida. Esta arquitectura se inspira del estudio previamente mencionado, donde se demostró que la combinación de enfoques convencionales y residuales mejora la precisión en la segmentación de estructuras complejas.

En la U-Net Doble, la salida de una U-Net convencional se utiliza como entrada para una U-Net Residual, aprovechando las ventajas de ambas arquitecturas. Esta configuración permite una mejor propagación de la información semántica a través de múltiples capas, lo que resulta en una segmentación más precisa y detallada. En la figura 31 se describe su arquitectura.

Figura 31

Arquitectura de U-Net Doble



Nota. Extraído de: (Quintana, Aceves, Villanueva, Tovar, 2021)

Los parámetros usados para entrenar el modelo de U-Net Doble se detallan en la tabla 20.

Tabla 20

Parámetros de entrenamiento de U-Net Doble

Parámetro	Valor
Tamaño de entrada	128 x 128 x 3
Optimizador	Adam (learning rate = 0.001)
Función de pérdida	Sparse Categorical Crossentropy
Tamaño de lote	32
Épocas	100
Callback	EarlyStopping ReduceLROnPlateau

Nota. Elaboración propia

En este caso, el total de parámetros entrenados fueron de 116,089,748 lo que equivale a un tamaño de 442.85 MB, siendo 38,696,582 (147.62 MB) los parámetros entrenados y 77,393,166 (295.23 MB) los parámetros del optimizador.

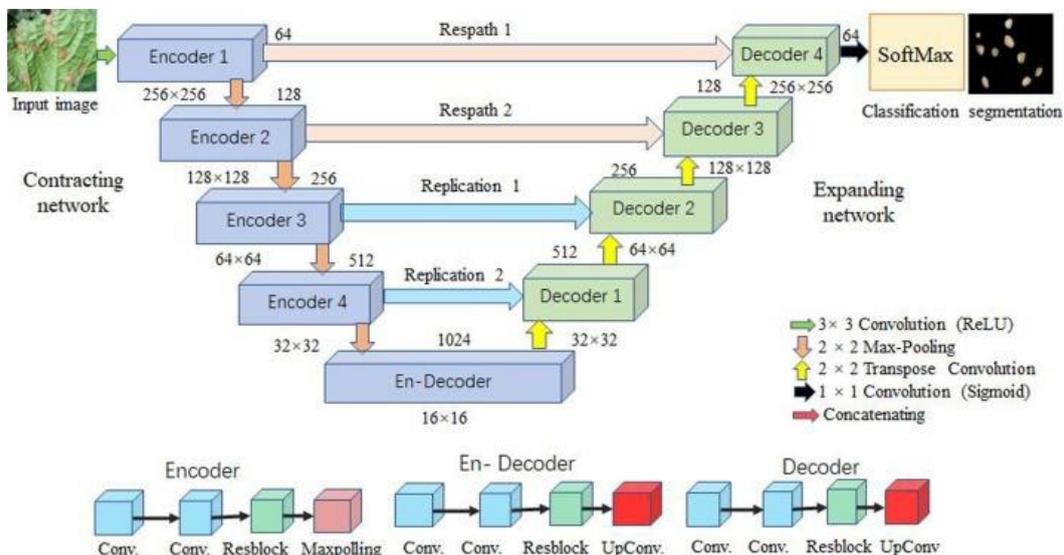
Modified U-Net

La Modified U-Net (MU-Net) es una adaptación de la arquitectura U-Net diseñada para mejorar la segmentación de imágenes complejas, como las hojas de cultivos enfermas. A diferencia de la U-Net tradicional, la MU-Net incorpora bloques residuales (Resblocks) y caminos residuales modificados (Respaths), los cuales ayudan a reducir la pérdida de información semántica entre las capas. Esta arquitectura fue inicialmente concebida para trabajar con imágenes de 256x256 píxeles; sin embargo, en este trabajo ha sido adaptada para procesar imágenes de 128x128 píxeles, las cuales corresponden al dataset utilizado en esta investigación.

La estructura principal de la MU-Net consiste en una red de codificación que extrae características clave de las imágenes y una red de decodificación que reconstruye la segmentación final a partir de esas características. El uso de DropBlock dentro de los Resblocks previene el sobreajuste, lo que es crucial para manejar la variabilidad y complejidad del dataset de hojas de cultivos. Además, las conexiones salteadas y los Respaths aseguran que la información espacial se mantenga intacta a lo largo de la red, mejorando así la precisión de la segmentación. En la figura 32 se detalla su arquitectura.

Figura 32

Arquitectura de Modified U-Net



Nota. Extraído de: (Shanwen Zhang, Chuanlei Zhang, 2023)

En la tabla 21 se detallan los parámetros usados para el entrenamiento de este modelo.

Tabla 21*Parámetros de entrenamiento de MU-Net*

Parámetro	Valor
Tamaño de entrada	128 x 128 x 3
Optimizador	Adam (learning rate = 0.001)
Función de pérdida	Sparse Categorical Crossentropy
Tamaño de lote	32
Épocas	100
Callback	ReduceLROnPlateau

Nota. Elaboración propia

El total de parámetros entrenados fueron 14,197,652 (54.16 MB), siendo 4,730,566 (18.05 MB) los parámetros entrenados, 5,952 (23.25 KB) los parámetros no entrenados y 9,461,134 (36.09 MB) los parámetros del optimizador.

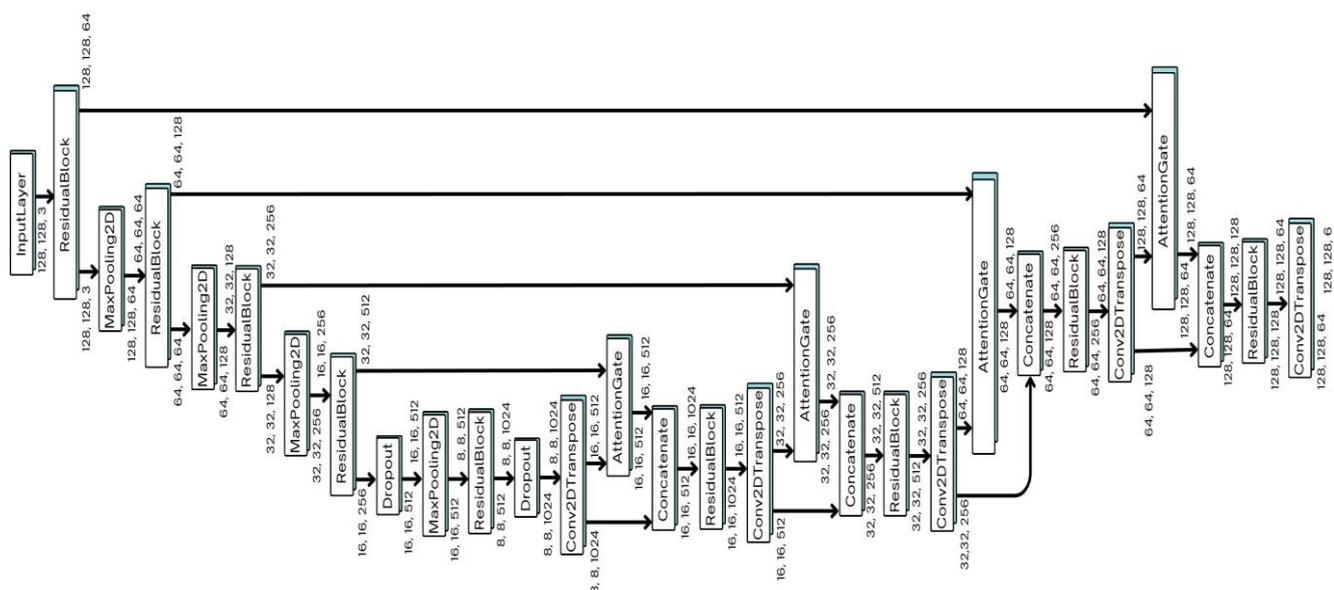
U-Net con módulo de atención y Resblock (U-NetAtt)

La U-Net con módulo de atención y Resblock es una adaptación del modelo U-Net original que mejora la segmentación de imágenes complejas. Su arquitectura incluye un encoder con capas de convolución y max-pooling que extraen características, seguidas de Resblocks que facilitan el flujo de gradientes, permitiendo que la red aprenda funciones de identidad y mejorando su capacidad de generalización. En el decoder, se integran módulos de atención que destacan regiones relevantes en las características extraídas, multiplicando un mapa de atención por las salidas del decoder antes de fusionarlas con las características del encoder. Esto ayuda a reducir la influencia de áreas irrelevantes y mejora la precisión de las segmentaciones finales.

Durante el entrenamiento, el modelo ajusta sus pesos para minimizar la pérdida entre las predicciones y las verdaderas máscaras de segmentación. La combinación de Resblocks y módulos de atención permite que el modelo retenga información espacial valiosa y se enfoque en las características relevantes, resultando en un rendimiento mejorado en tareas desafiantes de segmentación, como la identificación de malezas en imágenes agrícolas. En la figura 33 se describe su arquitectura.

Figura 33

Arquitectura de U-NetAtt



Nota. Elaboración propia

En la tabla 22 se detallan los parámetros usados para el entrenamiento del modelo.

Tabla 22

Parámetros de entrenamiento de U-NetAtt

Parámetro	Valor
Tamaño de entrada	128 x 128 x 3
Optimizador	Adam (learning rate = 0.001)
Función de pérdida	Sparse Categorical Crossentropy
Tamaño de lote	32
Épocas	100
Callback	ReduceLROnPlateau

Nota. Elaboración propia

El total de parámetros entrenados fueron 93,099,104 (355.14 MB), siendo 31,033,034 (118.38 MB) los parámetros entrenados y 62,066,070 (236.76 MB) los parámetros del optimizador.

2.8. Interpretación de datos

Después de realizar el entrenamiento del modelo de segmentación U-Net, se procedió

a su implementación y evaluación sobre nuevas imágenes. La interpretación de los datos y la utilización del modelo se llevaron a cabo a través de diversos métodos, los cuales se describen a continuación.

1. Carga y preparación de la imagen

Primero, se cargó el modelo entrenado desde su ubicación de almacenamiento y se preparó una imagen para la predicción. El modelo entrenado se cargó utilizando la función 'load_model' de la biblioteca Keras. Luego, la imagen a ser evaluada se cargó y se ajustó a las dimensiones requeridas por el modelo (128 x 128 píxeles en este caso).

2. Predicción

Una vez que la imagen fue adecuadamente preparada, se realizó la predicción utilizando el modelo cargado. El resultado de esta predicción es una matriz que contiene las probabilidades para cada clase en cada píxel de la imagen.

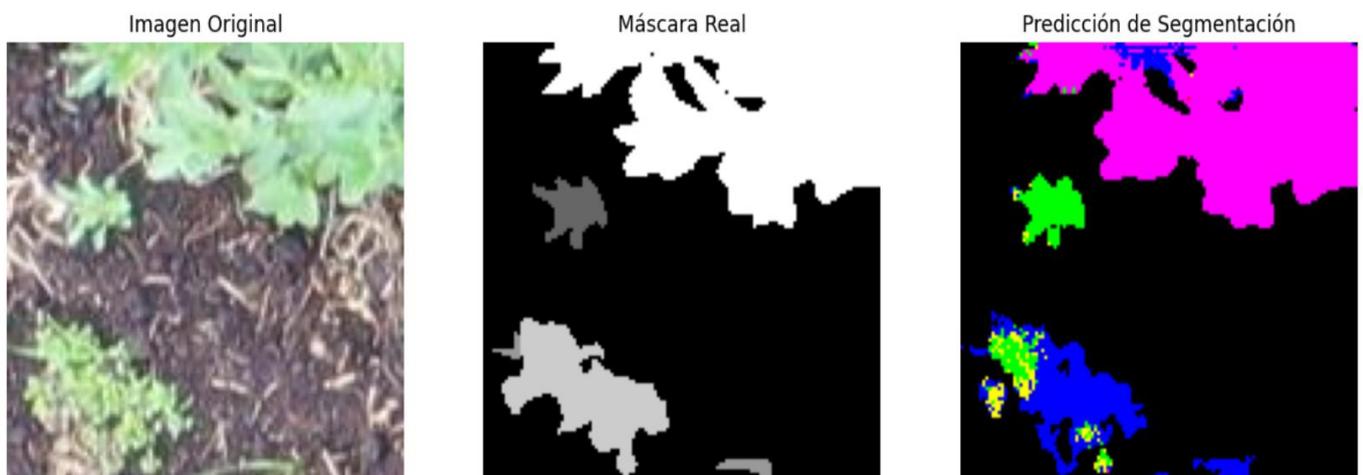
3. Visualización

Para una mejor comprensión de los resultados, se visualizó la segmentación de la imagen junto con la máscara real y la predicción generada por el modelo. Este proceso involucró la visualización de la imagen original, la carga y visualización de la máscara real asociada a la imagen original, y la conversión de la matriz de predicción en una imagen coloreada que representa las clases asignadas a cada píxel. Cada clase se representó con un color distinto para facilitar la interpretación.

En la figura 34 se muestra la imagen original, la máscara real y la predicción usando el modelo entrenado.

Figura 34

Visualización de segmentación en imagen de 128 x 128 píxeles



Nota. Elaboración propia

4. Cálculo del porcentaje de píxeles por clase

Para cuantificar la precisión de la segmentación, se calcularon los porcentajes de píxeles pertenecientes a cada clase tanto en la máscara real como en la predicción del modelo. El porcentaje de píxeles para una clase específica C_i se calcula usando la siguiente fórmula:

$$P(C_i) = \left(\frac{N(C_i)}{N_{total}} \right) \times 100$$

Donde:

- $P(C_i)$ es el porcentaje de píxeles de la clase C_i .
- $N(C_i)$ es el número de píxeles que pertenecen a la clase C_i .
- N_{total} es el número total de píxeles en la imagen.

De manera similar, se calculó el porcentaje de píxeles aplicado para la matriz de predicción del modelo usando la siguiente fórmula que toma como base la anterior:

$$P_{pred}(C_i) = \left(\frac{N_{pred}(C_i)}{N_{total}} \right) \times 100$$

Donde:

- $P_{pred}(C_i)$ es el porcentaje de píxeles predichos de la clase C_i .
- $N_{pred}(C_i)$ es el número de píxeles predichos que pertenecen a la clase C_i .
- N_{total} es el número total de píxeles en la predicción.

En la máscara real, se calculó el porcentaje de píxeles de cada clase, y de manera similar, se calculó el porcentaje de píxeles de cada clase en la predicción del modelo. Los resultados obtenidos en la imagen de muestra se describen en la tabla 23.

Tabla 23

Porcentaje de píxeles por clase en la máscara real y en la predicción

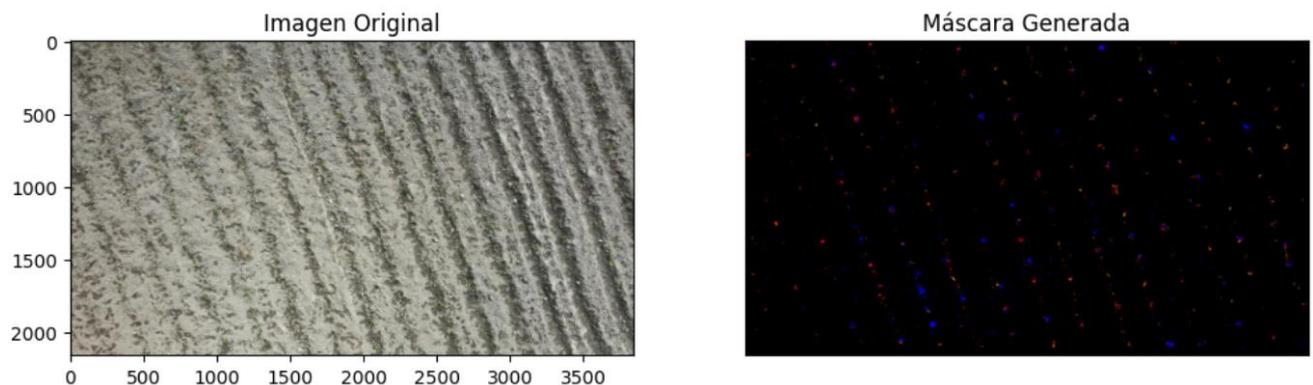
Clase	Máscara real (%)	Predicción (%)
Background	68.68%	67.99%
Cow-tongue	0.00%	0.01%
Dandelion	1.63%	2.64%
Kikuyo	0.56%	0.92%
Other	6.62%	6.54%
Potato	22.51%	21.91%

Nota. Elaboración propia

El modelo también se evaluó sobre imágenes completas, es decir, imágenes tomadas con el dron de 5472 x 3648 píxeles, que eran significativamente más grandes que las subimágenes de 128 x 128 píxeles utilizadas durante el entrenamiento. Para ello, se desarrolló una función que divide la imagen completa en subimágenes, realiza la predicción sobre cada una de ellas y luego reconstruye la predicción completa. Este método permite manejar imágenes de alta resolución sin comprometer la capacidad del modelo. Para esto se generó una máscara coloreada para visualizar la predicción de la imagen completa. Se utilizó un mapa de colores (colormap) para representar cada clase con un color distinto y se comparó la imagen original con la máscara generada para observar la precisión del modelo en la segmentación. En la figura 35 se visualiza el uso del modelo en una imagen completa.

Figura 35

Visualización de segmentación de imagen completa 5472 x 3648 píxeles



Nota. Elaboración propia

Para cuantificar el porcentaje de clases en imágenes completas se tomó como guía el método de cálculo usado para imágenes de 128 x 128 píxeles, adaptándolo para cualquier tipo de tamaño. Los resultados obtenidos se detallan en la tabla 24.

Tabla 24

Porcentaje de píxeles por clase en imagen completa 5472 x 3648 píxeles

Clase	Predicción (%)
Background	98.95%
Cow-tongue	0.44%
Dandelion	0.07%
Kikuyo	0.09%

Other	0.43%
Potato	0.02%

Nota. Elaboración propia

La interpretación de los datos obtenidos del modelo de segmentación U-Net mostró que el modelo es capaz de segmentar imágenes con un alto nivel de detalle y precisión. La comparación entre la máscara real y la predicción del modelo mediante visualizaciones y cálculos porcentuales permitió evaluar el rendimiento del modelo de manera efectiva. Además, el procesamiento de imágenes completas confirmó la capacidad del modelo para manejar tareas de segmentación en imágenes de alta resolución.

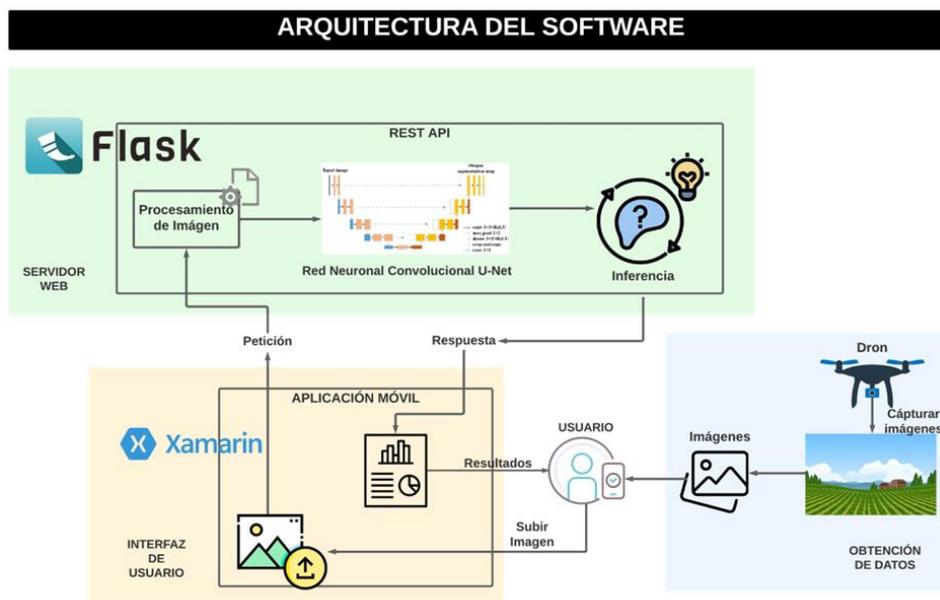
2.9. Despliegue del modelo

La última fase del proceso de descubrimiento de conocimiento en base de datos (KDD) es el despliegue del modelo. Esta fase implica la integración del modelo de aprendizaje automático en un entorno de producción donde puede ser utilizado para realizar predicciones en datos reales. El objetivo es proporcionar un sistema que permita a los usuarios interactuar con el modelo de manera eficiente y confiable. En este contexto, el despliegue implica no solo la implementación técnica del modelo, sino también la consideración de aspectos como la escalabilidad, la facilidad de uso y la mantenibilidad.

En fases anteriores del proyecto, se ha desarrollado un modelo de segmentación de imágenes utilizando la arquitectura U-Net. En esta fase se describe la integración del modelo en una aplicación móvil, permitiendo a los usuarios cargar imágenes y obtener predicciones a través de una interfaz intuitiva. La figura 36 ilustra la arquitectura general del sistema.

Figura 36

Arquitectura del sistema



Nota. Elaboración propia

2.9.1. Backend

La arquitectura del sistema incluye un backend desarrollado en Flask, un framework ligero para aplicaciones web en Python. La elección de Flask se debe a su simplicidad y flexibilidad, lo que facilita la creación de APIs REST eficientes, además con esta herramienta se pudo reutilizar gran parte de los métodos usados en la interpretación de datos realizados en Google Colab. El backend se desarrolló en Visual Studio Code y se ejecuta en un entorno Python versión 3.10.4. En esta sección se detallarán los componentes y funcionalidades.

Para su correcto funcionamiento fue necesario instalar ciertas dependencias que se describen en la tabla 25.

Tabla 25

Dependencias instaladas

Librería	Versión
Flask	2.0.3
Werkzeug	2.0.3
TensorFlow	2.8.0
Protobuf	3.20.*
Pillow	10.4.0
NumPy	2.0.0
Matplotlib	3.9.1

Nota. Elaboración propia

Estas bibliotecas son esenciales para manejar solicitudes web, procesar imágenes y ejecutar el modelo de aprendizaje profundo.

2.9.1.1. Flujo de trabajo de predicción

El código del backend define varias rutas y funciones clave. La ruta principal es '/predict', que maneja las solicitudes POST para la predicción de segmentación. Al recibir una solicitud con una imagen, la aplicación realiza los pasos detallados en la figura 37.

Figura 37

Flujo de trabajo de predicción del backend



Nota. Elaboración propia

Primero la imagen proporcionada se carga y se convierte en un array NumPy, preparándola para ser procesada por el modelo. Se divide la imagen en subimágenes de tamaño 128 x 128 píxeles, con un paso de superposición de 64 píxeles para asegurar que se cubran todas las áreas. Cada subimagen se pasa a través del modelo U-Net cargado para obtener las predicciones. Estas predicciones se ensamblan para formar una predicción completa para la imagen. La predicción resultante se convierte en una máscara coloreada usando un mapa de colores predefinido. La misma que se guarda como una imagen en formato PNG y se codifica en base64 para facilitar su transmisión a través de la API. Finalmente, se calcula el porcentaje de píxeles que pertenecen a cada clase en la imagen segmentada. Estos porcentajes proporcionan información útil sobre la distribución de las clases en la imagen y son devueltos en formato JSON juntamente con la imagen en base64 como respuesta a la petición.

Además, se definió la ruta '/hello' como un punto de prueba simple que devuelve un saludo, confirmando que el servidor está funcionando correctamente.

2.9.2. Frontend

El desarrollo del Frontend de la aplicación se llevó a cabo utilizando Visual Studio 2022 con Xamarin, una plataforma que permite crear aplicaciones móviles para Android e iOS con un solo código base en C#. Esta decisión fue tomada debido a la capacidad de Xamarin de integrarse fácilmente con diversas bibliotecas y herramientas, y su fuerte soporte para el desarrollo multiplataforma. El archivo del proyecto en Visual Studio 2022 incluye referencias a varios paquetes de NuGet que son esenciales para el funcionamiento de la aplicación. Estos paquetes proporcionan funcionalidades adicionales y optimizan el desarrollo de la aplicación. Los paquetes usados incluyen:

- **Newtonsoft.Json:** Esta biblioteca se utiliza para la serialización y deserialización de datos JSON, facilitando la comunicación con la API Flask.
- **Plugin.Permissions:** Gestiona los permisos necesarios en dispositivos móviles, como acceso a la cámara y almacenamiento.
- **Rg.Plugins.Popup:** Permite la creación de ventanas emergentes, mejorando la interacción del usuario con la aplicación.
- **SkiaSharp** y **SkiaSharp.Views.Forms:** Utilizadas para renderizar gráficos y proporcionar capacidades de dibujo avanzadas dentro de la aplicación.
- **Xam.Plugin.Media:** Facilita el acceso a la cámara y la galería de fotos del dispositivo para capturar imágenes.
- **Xanimation:** Proporciona animaciones para mejorar la experiencia del usuario.
- **Xamarin.Forms:** El marco principal para construir interfaces de usuario.
- **Xamarin.Essentials:** Ofrece APIs multiplataforma para las características más comunes de dispositivos móviles.

En la tabla 26 se resume los paquetes NuGet usados en el proyecto.

Tabla 26

Paquetes NuGet instalados

Paquete	Versión	Funcionalidad
Newtonsoft.Json	13.0.3	Serialización y deserialización de JSON
Plugin.Permissions	6.0.1	Gestión de permisos
Rg.Plugins.Popup	2.1.0	Ventanas emergentes
SkiaSharp	2.88.8	Renderización de gráficos
Xam.Plugin.Media	6.0.2	Acceso a cámara y galería

Xanimation	1.3.0	Animaciones
Xamarin.Forms	5.0.0.2196	Construcción de interfaces de usuario
Xamarin.Essentials	1.7.0	APIs multiplataforma

Nota. Elaboración propia

El Frontend de la aplicación consta de dos pantallas principales, la primera es la pantalla de bienvenida, siendo la interfaz inicial que se muestra al usuario cuando abre la aplicación. Asimismo, está la página de inferencias en donde se lleva a cabo todo el proceso de interacción con la API Flask. En la figura 38 y 39 se visualiza la pantalla de bienvenida y la de inferencias respectivamente.

Figura 38

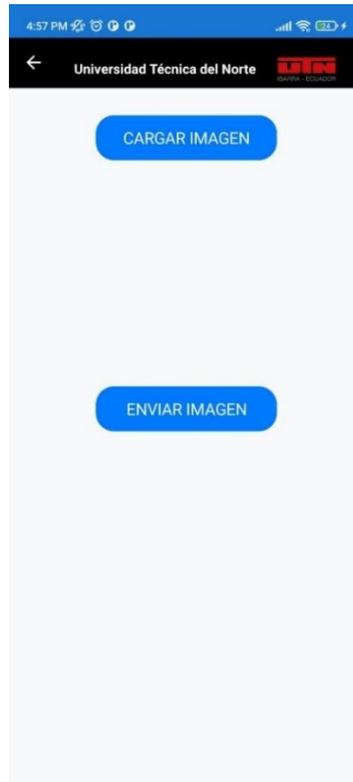
Pantalla de bienvenida



Nota. Elaboración propia

Figura 39

Pantalla de inferencias



Nota. Elaboración propia

2.9.2.1. Consumo de la API Flask

En el Frontend se consume la API Flask desarrollada para la segmentación de imágenes y la cuantificación de malezas. La API recibe una imagen y devuelve una máscara de segmentación junto con los porcentajes de píxeles por clase. A continuación, se presenta el flujo de trabajo para realizar esta interacción:

1. Captura o selección de Imagen

El usuario puede capturar una imagen usando la cámara del dispositivo o seleccionar una imagen existente desde la galería.

2. Envío de imagen a la API

La imagen seleccionada se envía a la API Flask a través de una solicitud HTTP POST. La biblioteca 'HttpClient' en Xamarin se utiliza para manejar esta solicitud.

3. Procesamiento de la respuesta

La respuesta de la API, que incluye la máscara de segmentación y los porcentajes de clases, se recibe y se procesa. La imagen de la máscara se muestra al usuario junto con un

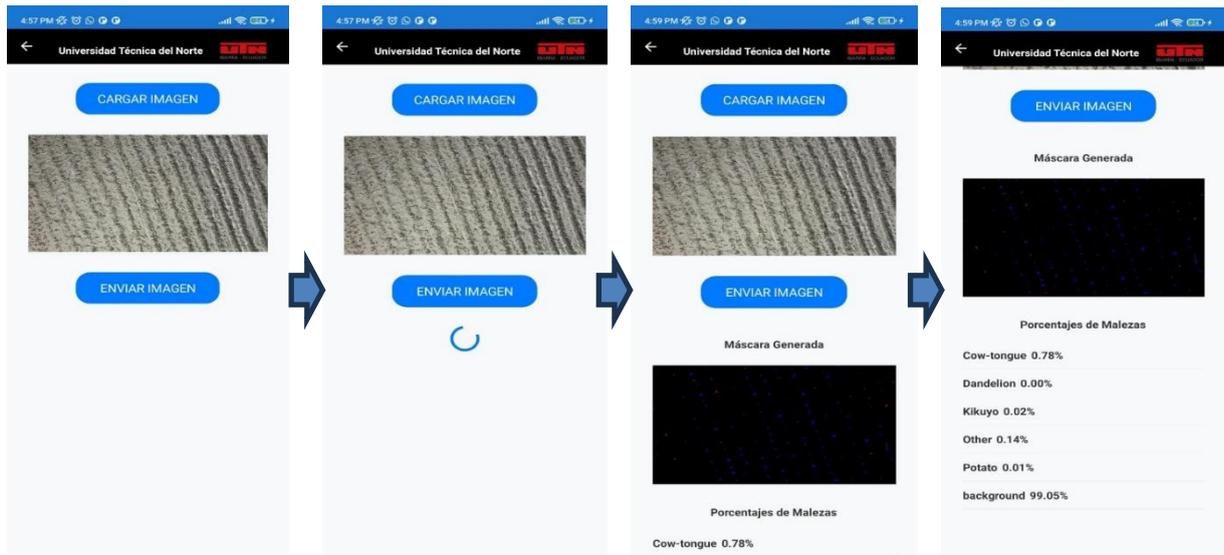
análisis detallado de los porcentajes de malezas.

4. Visualización de resultados

Los resultados se muestran de manera intuitiva una vez acabado este proceso. En la figura 40 se muestra el proceso que sigue la aplicación.

Figura 40

Flujo de trabajo de la pantalla de inferencia



Nota. Elaboración propia

2.9.3. Dockerización

Para garantizar una instalación y despliegue eficiente en diferentes entornos, se optó por dockerizar la aplicación. Docker permite crear contenedores que encapsulan todas las dependencias y configuraciones necesarias, asegurando que la aplicación se ejecute de manera consistente independientemente del entorno en el que se despliegue.

La dockerización del proyecto implica la creación de un archivo Dockerfile que define el entorno de ejecución de la aplicación. Este archivo incluye instrucciones para instalar todas las dependencias necesarias y copiar el código de la aplicación al contenedor. Al construir la imagen de Docker y ejecutar un contenedor basado en esta imagen, se garantiza que la aplicación se ejecutará en un entorno controlado y replicable.

CAPÍTULO 3

Validación de resultados

La validación de resultados es un proceso para evaluar la efectividad y eficiencia tanto del modelo U-Net implementado como la solución tecnológica propuesta en este trabajo de investigación. Se realizarán tres tipos de validación, la primera es una comparativa de resultados de entrenamientos de las versiones del modelo de red neuronal convolucional U-Net, la segunda es la validación según la norma ISO 25023 enfocada a la característica del comportamiento del tiempo, usando la métrica de tiempo de respuesta del software propuesto. Finalmente, la tercera validación es la comparación de tiempos entre el método manual y el método automatizado propuesto para realizar la tarea de identificación y cuantificación de malezas. Cada una de estas validaciones se describe a continuación:

3.1. Comparativa de resultados de entrenamientos del modelo U-Net

La primera validación se centra en la comparación de los resultados obtenidos a partir de los diferentes entrenamientos del modelo U-Net. Para ello, se utilizarán métricas de inteligencia artificial ampliamente reconocidas en la comunidad científica. Las métricas seleccionadas incluyen el dice loss, mean dice coefficient y el índice de Jaccard (Intersection over Union, IoU).

El objetivo de esta validación es evaluar y comparar el rendimiento de las diferentes versiones del modelo U-Net entrenadas con distintas configuraciones de datos y parámetros. Se espera que las métricas de rendimiento proporcionen una visión clara de la capacidad del modelo para segmentar correctamente las malezas en las imágenes de entrada y así poder elegir el mejor modelo para trabajar.

En la tabla 27 se describe las métricas de evaluación del modelo U-Net.

Tabla 27

Métricas de evaluación del modelo U-Net

Métrica	Descripción	Fórmula
Dice loss	Evalúa el nivel de superposición y semejanza que existe entre la máscara de segmentación de la predicción y la máscara verdadera.	$1 - \frac{1}{C} \sum_{c=1}^C \frac{2 \sum_i^N Y_{pred_{i,c}} Y_{true_{i,c}}}{\sum_i^N Y_{pred_{i,c}}^2 + \sum_i^N Y_{true_{i,c}}^2}$

Mean dice coefficient	Es el valor promedio del coeficiente Dice para todas las clases. Cuanto más cerca esté de 1, mejor será la segmentación.	$\frac{1}{C} \sum_{c=1}^C \frac{2 \sum_i^N Y_{pred_{i,c}} Y_{true_{i,c}}}{\sum_i^N Y_{pred_{i,c}}^2 + \sum_i^N Y_{true_{i,c}}^2}$
Índice de Jaccard (mean IoU)	Proporción de la intersección sobre la unión de los conjuntos	$\frac{1}{C} \sum_{c=1}^C \frac{\sum_i^N Y_{pred_{i,c}} \cap Y_{true_{i,c}}}{\sum_i^N Y_{pred_{i,c}} \cup Y_{true_{i,c}}}$

Nota. Elaboración propia

Donde:

- **Y pred** es la máscara predicha por la capa softmax en codificación one-hot.
- **Y true** es la máscara real, ground truth en codificación one-hot.
- **C** es el número de clases.
- **N** es el número de píxeles.
- **i** es el inicio del intervalo hasta N, su valor inicial es 1.

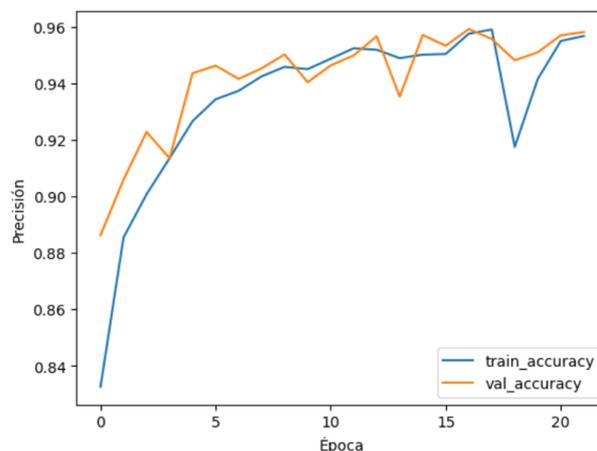
La comparación de resultados fue entre las diferentes versiones implementadas en el capítulo 2. A continuación se detallan los resultados obtenidos por cada modelo.

U-Net Base Line

En la figura 41 y 42 se observa los resultados obtenidos con respecto a la precisión y perdida respectivamente.

Figura 41

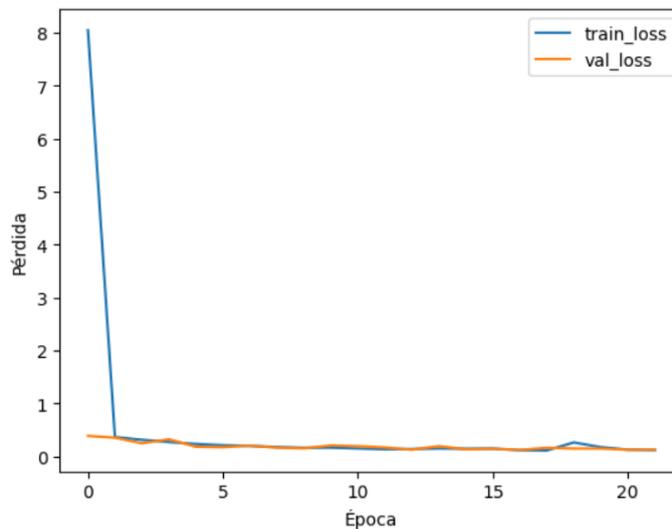
Gráfico de precisión U-Net



Nota. Elaboración propia

Figura 42

Gráfico de pérdida U-Net

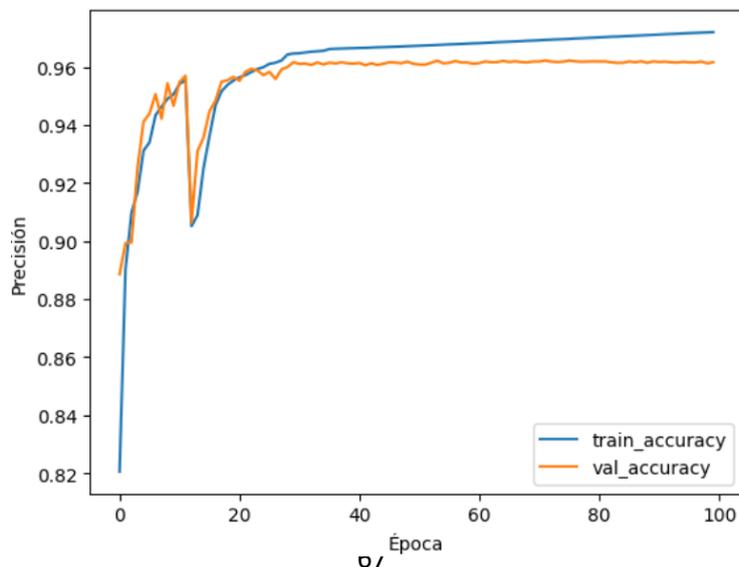


Nota. Elaboración propia

Los gráficos anteriores muestran la evolución de la pérdida y la precisión durante el entrenamiento del modelo. Se puede observar que el modelo alcanzó una alta precisión y una baja pérdida en el conjunto de validación, indicando un buen rendimiento en la tarea de segmentación. Sin embargo, en ambas figuras se puede notar que el entrenamiento no completó las 100 épocas establecidas, esto debido al callback EarlyStopping que detuvo el proceso al no haber mejoría en los resultados. En la figura 43 y 44 se muestra los resultados que se obtienen al no hacer uso del callback EarlyStopping.

Figura 43

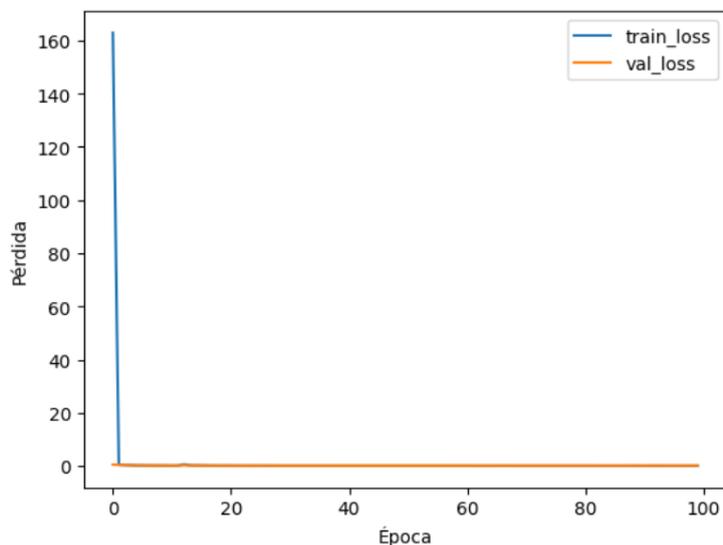
Gráfico de precisión U-Net 100 épocas



Nota. Elaboración propia

Figura 44

Gráfico de pérdida U-Net 100 épocas



Nota. Elaboración propia

Con este enfoque observamos una mejoría en los resultados de precisión y pérdida en los conjuntos de entrenamiento y validación a diferencia de los anteriores usando el callback EarlyStopping.

En la tabla 28 se exponen los resultados con respecto a las métricas mean dice coefficient, dice loss y el índice de Jaccard o mean IoU en el conjunto de entrenamiento y validación. Adicionalmente, en la tabla 29 se muestra el mean IoU para cada clase utilizando el conjunto de validación.

Tabla 28

Resultados mean dice coefficient, dice loss y mean IoU de U-Net

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.8673	0.7833	0.1326
Validación	0.7575	0.6541	0.2425

Nota. Elaboración propia

Tabla 29*Resultados mean IoU para cada clase U-Net*

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9693	0.6308	0.5826	0.6546	0.1832	0.9040	0.6541

Nota. Elaboración propia

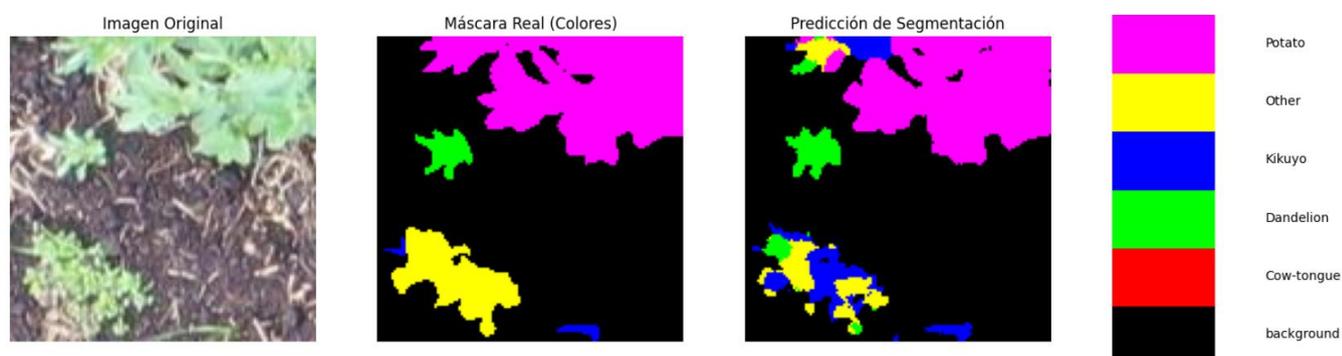
El tiempo de inferencia para una imagen de 128×128 píxeles y para una imagen completa capturada por el dron de 5472×3648 píxeles se detallan en la tabla 30.

Tabla 30*Tiempos de inferencia de U-Net*

Tipo de imagen	Tiempo (segundos)
Imagen de 128×128 px	0.0613
Imagen capturada por el dron de 5472×3648 px	120.02
Entrenamiento 100 épocas	8700

Nota. Elaboración propia

En la figura 45 se visualiza el resultado de predicción y la máscara real de una imagen de 128×128 píxeles.

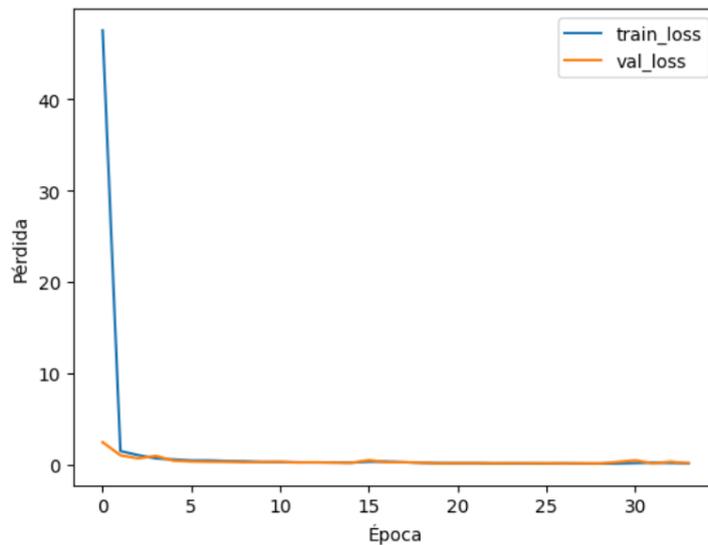
Figura 45*Resultado de predicción y máscara real con U-Net**Nota.* Elaboración propia

U-Net Residual

Los resultados de pérdida y precisión dentro del conjunto de datos de entrenamiento y el de validación se muestra en la figura 46 y 47 respectivamente.

Figura 46

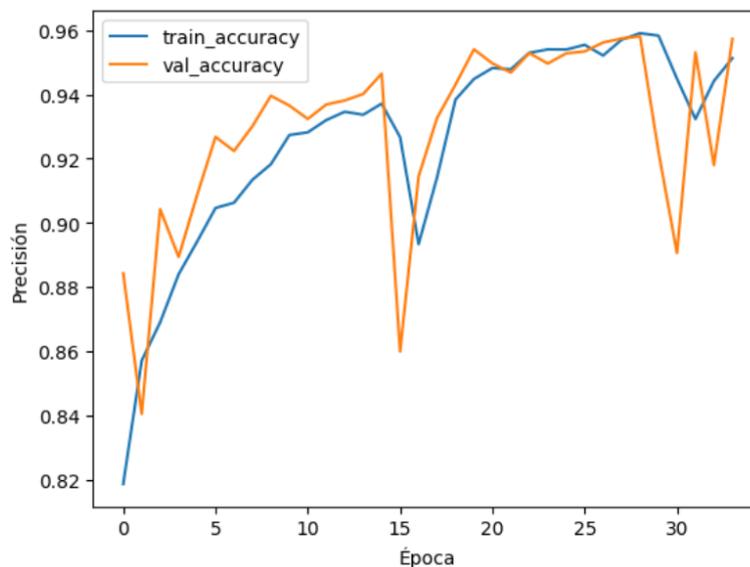
Gráfico de pérdida de U-Net Residual



Nota. Elaboración propia

Figura 47

Gráfico de precisión de U-Net Residual



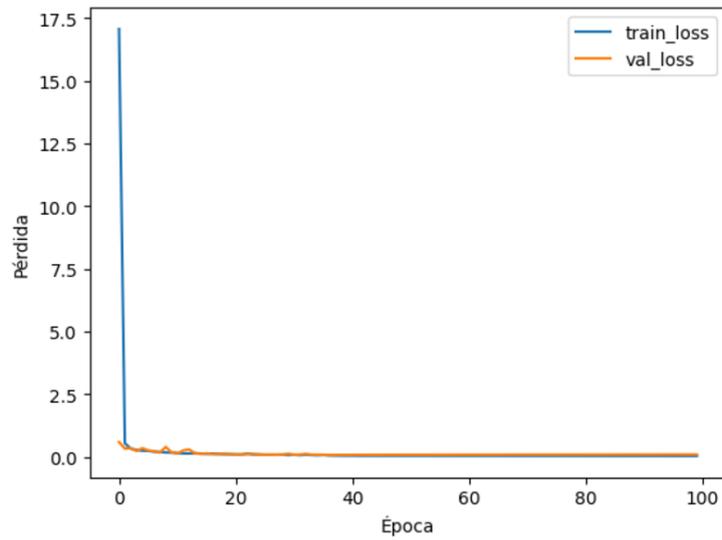
Nota. Elaboración propia

Inicialmente, se utilizó el callback EarlyStopping, lo que resultó en una finalización anticipada del entrenamiento. Para evaluar el impacto de entrenamientos más prolongados y

analizar la convergencia del modelo, se realizó un segundo experimento excluyendo el callback EarlyStopping y empleando únicamente el callback ReduceLROnPlateau durante 100 épocas completas. A continuación, se detallan los resultados en la figura 48 y 49.

Figura 48

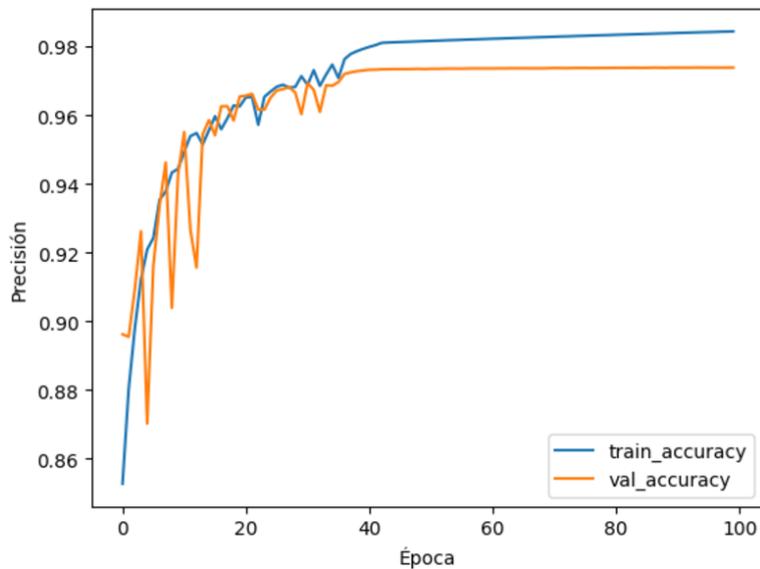
Gráfico de pérdida de U-Net Residual 100 épocas



Nota. Elaboración propia

Figura 49

Gráfico de precisión de U-Net Residual 100 épocas



Nota. Elaboración propia

En la tabla 31 se exponen los resultados con respecto a las métricas mean dice coefficient, dice loss y el índice de Jaccard o mean IoU en el conjunto de entrenamiento y

validación con el dataset desbalanceado. En la tabla 33 se muestra el mean IoU para cada clase utilizando el conjunto de validación. Adicionalmente, en la tabla 32 se expone los resultados con respecto a las métricas usadas anteriormente en el conjunto de entrenamiento, validación y test con el dataset balanceado. En la tabla 34 se muestra el mean IoU para cada clase utilizando el conjunto de prueba.

Tabla 31

Resultados mean dice coefficient, dice loss y mean IoU de U-Net Residual (dataset desbalanceado)

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.9865	0.9734	0.0135
Validación	0.8235	0.7755	0.1765

Nota. Elaboración propia

Tabla 32

Resultados mean dice coefficient, dice loss y mean IoU de U-Net Residual (dataset balanceado)

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.9521	0.9096	0.0478
Validación	0.8619	0.7736	0.1380
Test	0.8763	0.8021	0.1236

Nota. Elaboración propia

Tabla 33

Resultados mean IoU para cada clase U-Net Residual (dataset desbalanceado)

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9676	0.7626	0.7404	0.7053	0.5564	0.9204	0.7755

Nota. Elaboración propia

Tabla 34*Resultados mean IoU para cada clase U-Net Residual (dataset balanceado)*

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9721	0.7984	0.7409	0.7730	0.5907	0.9376	0.8021

Nota. Elaboración propia

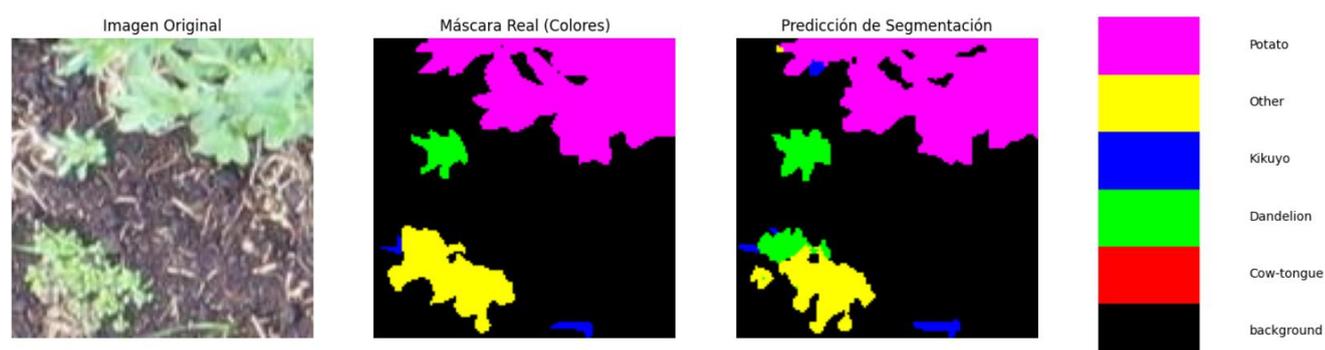
El tiempo de inferencia para una imagen de 128×128 píxeles y para una imagen completa capturada por el dron de 5472×3648 píxeles se detallan en la tabla 35.

Tabla 35*Tiempos de inferencia de U-Net Residual*

Tipo de imagen	Tiempo (segundos)
Imagen de 128×128 px	0.0585
Imagen capturada por el dron de 5472×3648 px	122.01
Entrenamiento 100 épocas	9000

Nota. Elaboración propia

En la figura 50 se visualiza el resultado de predicción y la máscara real de una imagen de 128×128 píxeles.

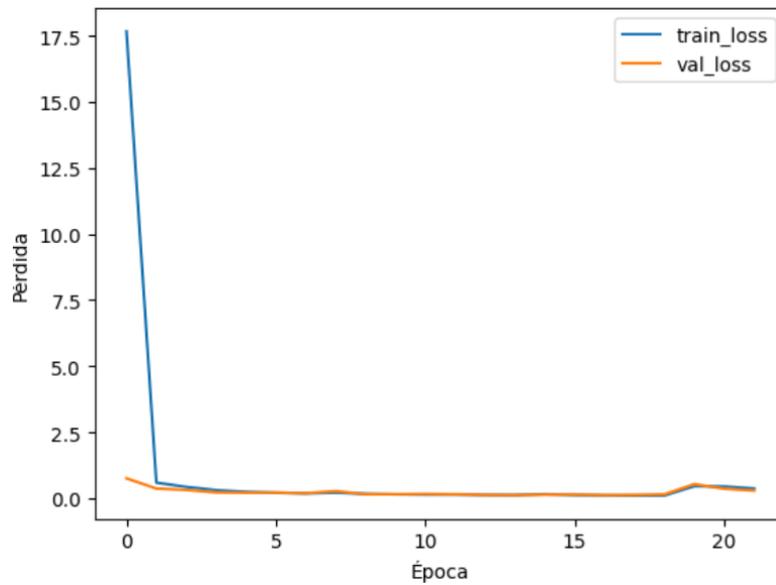
Figura 50*Resultado de predicción y máscara real con U-Net Residual**Nota.* Elaboración propia

U-Net Doble

Los resultados de pérdida y precisión dentro del conjunto de datos de entrenamiento y el de validación se muestra en la figura 51 y 52 respectivamente.

Figura 51

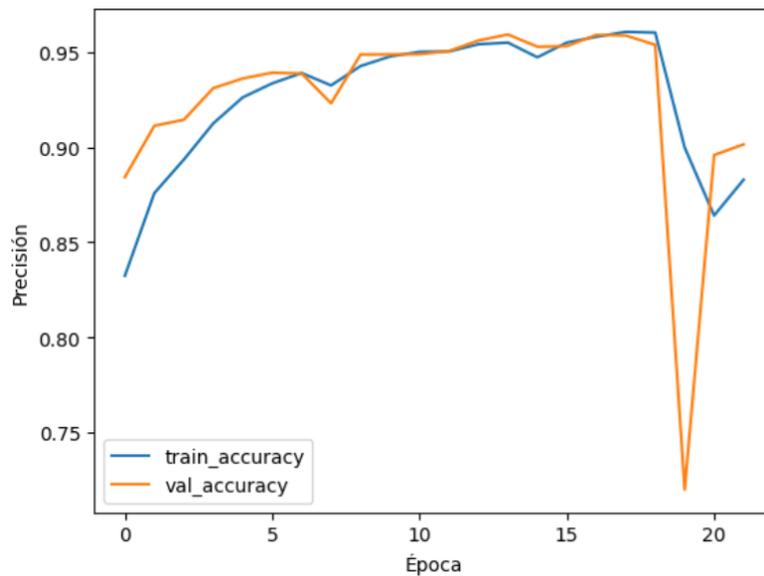
Gráfico de pérdida de U-Net Doble



Nota. Elaboración propia

Figura 52

Gráfico de precisión de U-Net Doble



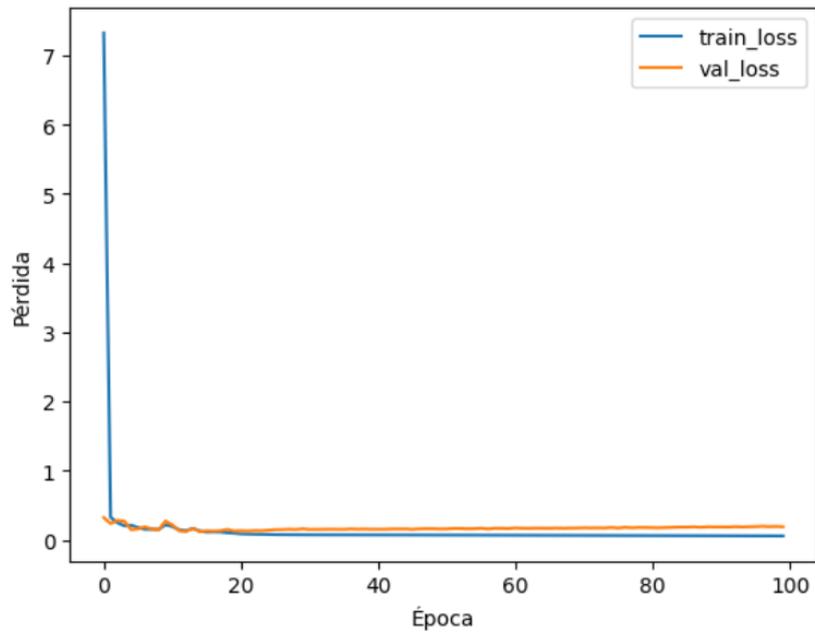
Nota. Elaboración propia

Al igual que con los anteriores modelos, se llevó a cabo un experimento inicial

utilizando el callback EarlyStopping, seguido de un segundo experimento en el que se entrenó el modelo durante las 100 épocas completas utilizando solo el callback ReduceLROnPlateau. A continuación, se presentan los resultados en la figura 53 y 54.

Figura 53

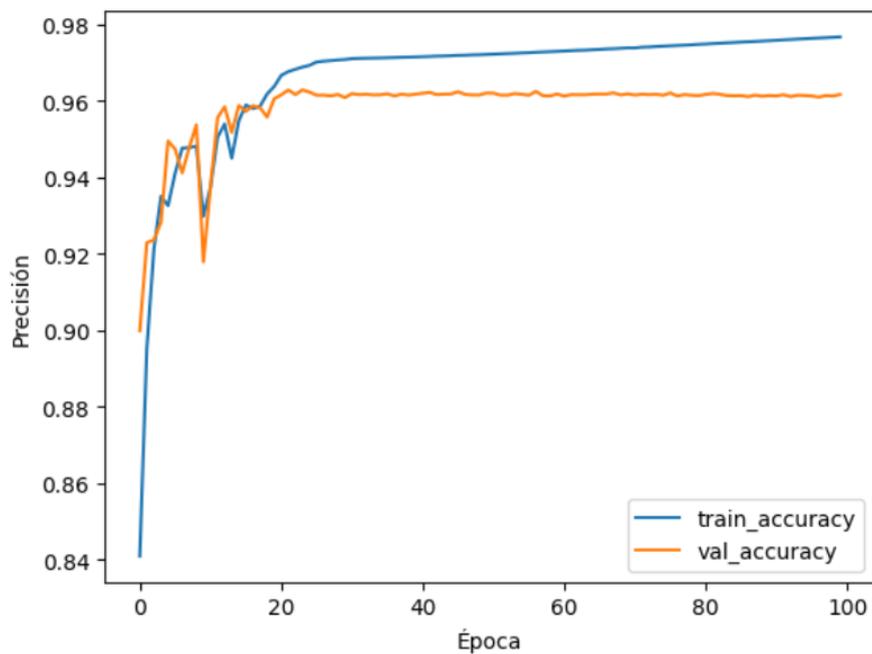
Gráfico de pérdida de U-Net Doble 100 épocas



Nota. Elaboración propia

Figura 54

Gráfico de precisión de U-Net Doble 100 épocas



Nota. Elaboración propia

En la tabla 36 se exponen los resultados con respecto a las métricas mean dice coefficient, dice loss y el índice de Jaccard o mean IoU en el conjunto de entrenamiento y validación. Adicionalmente, en la tabla 37 se muestra el mean IoU para cada clase utilizando el conjunto de validación.

Tabla 36

Resultados mean dice coefficient, dice loss y mean IoU de U-Net Doble

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.9364	0.8827	0.0636
Validación	0.7530	0.6488	0.2470

Nota. Elaboración propia

Tabla 37

Resultados mean IoU para cada clase U-Net Doble

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9659	0.6812	0.5648	0.6061	0.1782	0.8965	0.6488

Nota. Elaboración propia

El tiempo de inferencia para una imagen de 128×128 píxeles y para una imagen completa capturada por el dron de 5472×3648 píxeles se detallan en la tabla 38.

Tabla 38

Tiempos de inferencia de U-Net Doble

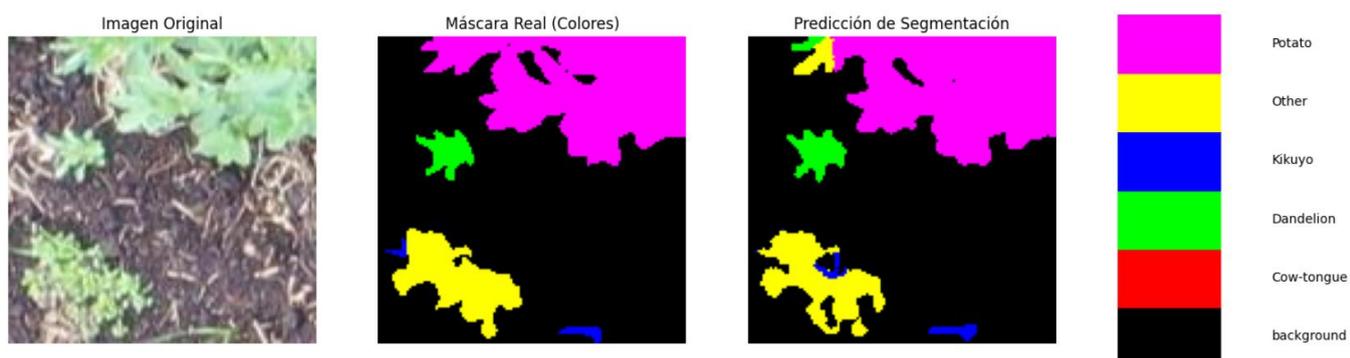
Tipo de imagen	Tiempo (segundos)
Imagen de 128×128 px	0.0583
Imagen capturada por el dron de 5472×3648 px	121.83
Entrenamiento 100 épocas	9600

Nota. Elaboración propia

En la figura 55 se visualiza el resultado de predicción y la máscara real de una imagen de 128x128 píxeles.

Figura 55

Resultado de predicción y máscara real con U-Net Doble



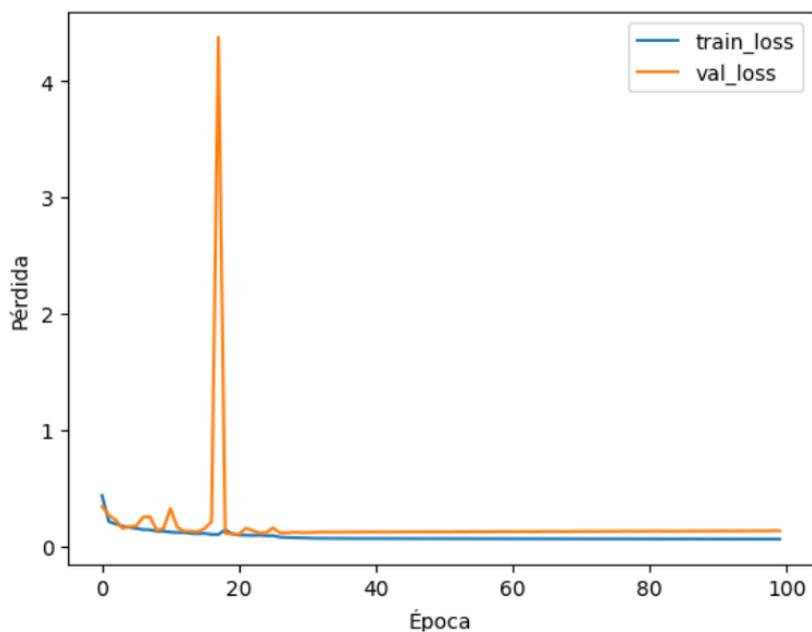
Nota. Elaboración propia

MU-Net

Los resultados en el conjunto de entrenamiento y validación se observan en las figuras 56 y 57.

Figura 56

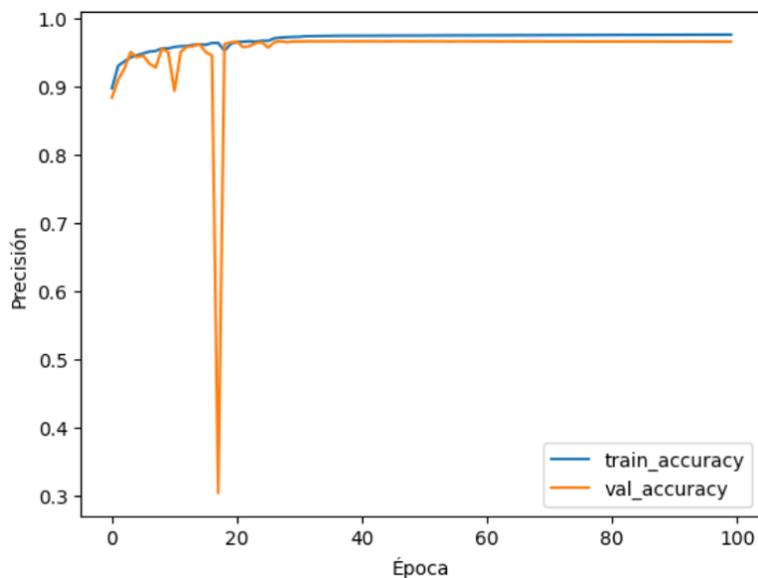
Gráfico de pérdida de MU-Net



Nota. Elaboración propia

Figura 57

Gráfico de precisión de MU-Net



Nota. Elaboración propia

A diferencia de las anteriores versiones, se decidió entrenar el modelo sin el callback EarlyStopping y culminar las 100 épocas.

En la tabla 39 se exponen los resultados con respecto a las métricas mean dice coefficient, dice loss y el índice de Jaccard o mean IoU en el conjunto de entrenamiento y validación. Adicionalmente, en la tabla 40 se muestra el mean IoU para cada clase utilizando el conjunto de validación.

Tabla 39

Resultados mean dice coefficient, dice loss y mean IoU de MU-Net

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.9149	0.8476	0.0850
Validación	0.7764	0.6790	0.2235

Nota. Elaboración propia

Tabla 40*Resultados mean IoU para cada clase MU-Net*

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9706	0.7463	0.5627	0.6763	0.2010	0.9174	0.6790

Nota. Elaboración propia

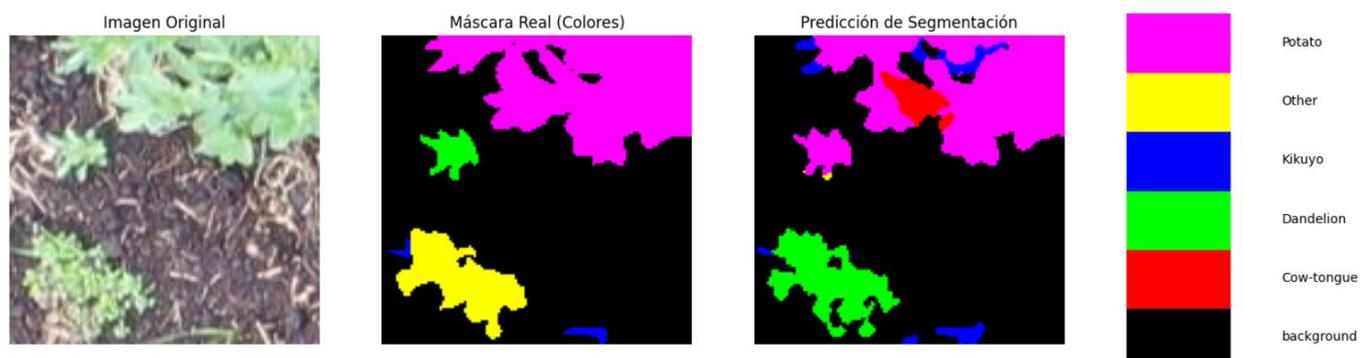
El tiempo de inferencia para una imagen de 128×128 píxeles y para una imagen completa capturada por el dron de 5472×3648 píxeles se detallan en la tabla 41.

Tabla 41*Tiempos de inferencia de MU-Net*

Tipo de imagen	Tiempo (segundos)
Imagen de 128×128 px	0.0587
Imagen capturada por el dron de 5472×3648 px	128.17
Entrenamiento 100 épocas	10200

Nota. Elaboración propia

En la figura 58 se visualiza el resultado de predicción y la máscara real de una imagen de 128×128 píxeles.

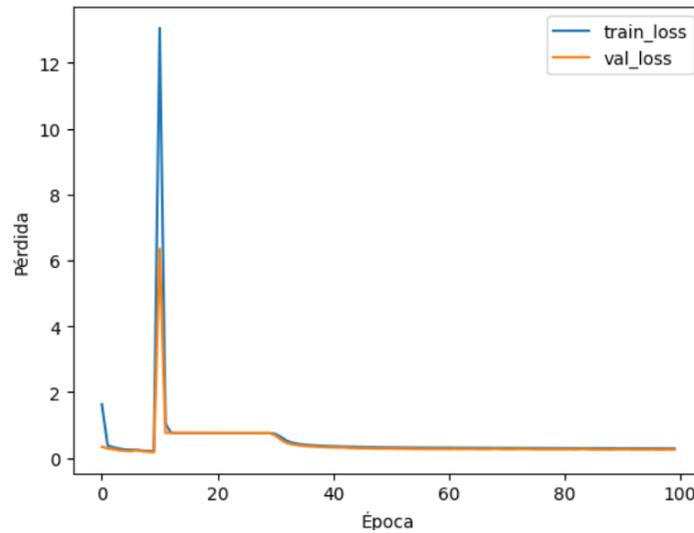
Figura 58*Resultado de predicción y máscara real con MU-Net**Nota.* Elaboración propia

U-NetAtt

Los resultados de precisión y pérdida en el conjunto de entrenamiento y validación se observan en las figuras 59 y 60.

Figura 59

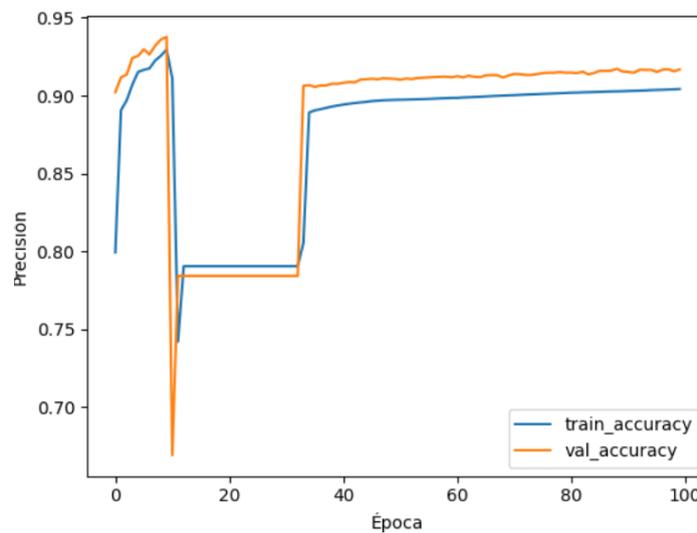
Gráfico de pérdida de U-NetAtt



Nota. Elaboración propia

Figura 60

Gráfico de precisión de U-NetAtt



Nota. Elaboración propia

A diferencia de las anteriores versiones, se decidió entrenar el modelo sin el callback EarlyStopping y culminar las 100 épocas.

En la tabla 42 se exponen los resultados con respecto a las métricas mean dice

coefficient, dice loss y el índice de Jaccard o mean IoU en el conjunto de entrenamiento y validación. Adicionalmente, en la tabla 43 se muestra el mean IoU para cada clase utilizando el conjunto de validación.

Tabla 42

Resultados mean dice coefficient, dice loss y mean IoU de U-NetAtt

Conjunto	Mean Dice Coefficient	Mean IoU	Dice Loss
Entrenamiento	0.8644	0.8109	0.1355
Validación	0.6647	0.6150	0.3352

Nota. Elaboración propia

Tabla 43

Resultados mean IoU para cada clase U-NetAtt

IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
0.9600	0.7122	0.4811	0.6819	0.1274	0.7271	0.6150

Nota. Elaboración propia

El tiempo de inferencia para una imagen de 128×128 píxeles y para una imagen completa capturada por el dron de 5472×3648 píxeles se detallan en la tabla 44.

Tabla 44

Tiempos de inferencia de U-NetAtt

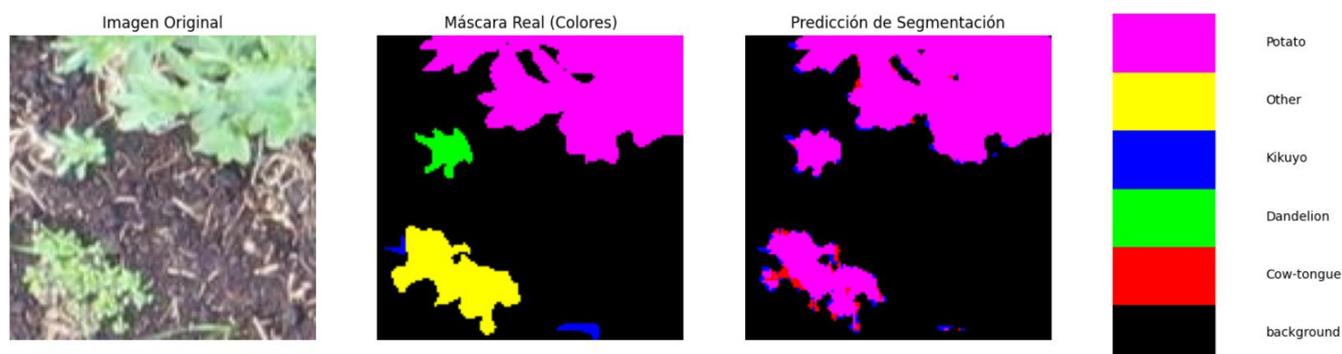
Tipo de imagen	Tiempo (segundos)
Imagen de 128×128 px	0.0570
Imagen capturada por el dron de 5472×3648 px	122.02
Entrenamiento 100 épocas	8100

Nota. Elaboración propia

En la figura 61 se visualiza el resultado de predicción y la máscara real de una imagen de 128x128 píxeles.

Figura 61

Resultado de predicción y máscara real con U-NetAtt



Nota. Elaboración propia

3.1.1. Comparación en conjunto

En la tabla 45 se detallan los resultados obtenidos con respecto a la precisión y pérdida de los modelos U-Net original, U-Net Residual, U-Net Doble, MU-Net y U-NetAtt en el conjunto de datos de entrenamiento y el conjunto de validación. En la figura 62 se muestra las respectivas gráficas en ambos conjuntos de todos los modelos tanto de precisión como de pérdida.

Tabla 45

Comparativa de resultados en el conjunto de entrenamiento y validación

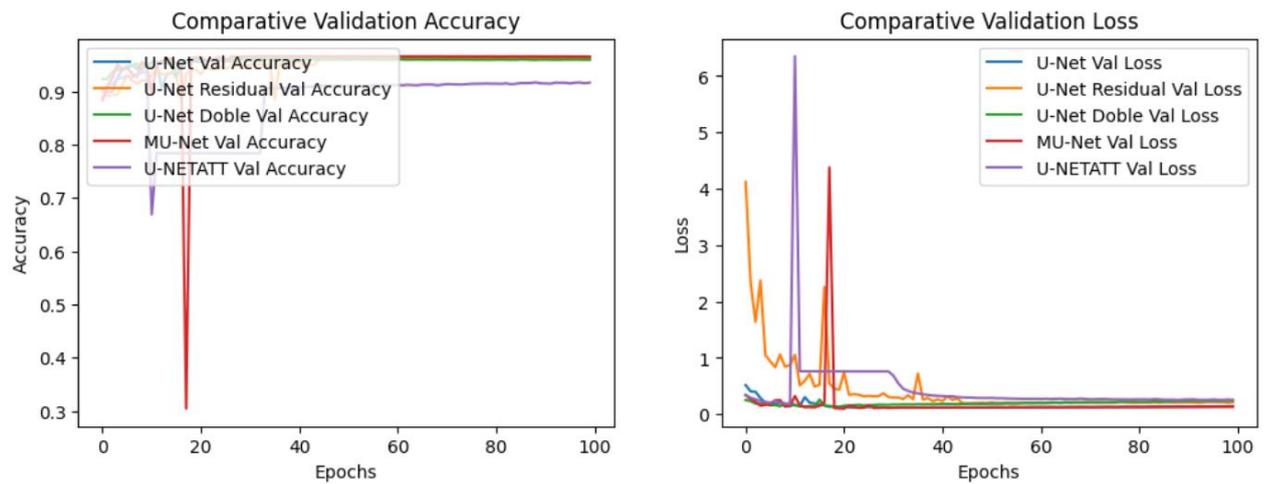
Modelo	Precisión entrenamiento	Precisión validación	Pérdida entrenamiento	Pérdida validación
U-Net Base Line	0.9568	0.9582	0.1209	0.1282
U-Net 100 épocas	0.9720	0.9616	0.0721	0.1424
U-Net Residual	0.9581	0.9573	0.1443	0.1305
U-Net Residual 100 épocas	0.9954	0.9647	0.0115	0.3862
U-Net Doble	0.9607	0.9588	0.1064	0.1299
U-Net Doble 100 épocas	0.9806	0.9597	0.0469	0.2414

MU-Net	0.9758	0.9655	0.0597	0.1305
U-NetAtt	0.9042	0.9168	0.2877	0.2549

Nota. Elaboración propia

Figura 62

Gráficas de comparación de precisión y pérdida



Nota. Elaboración propia

En la tabla 46 se describen los resultados con respecto a las métricas: dice loss, mean dice coefficient y el índice de Jaccard (mean IoU) de las mejores versiones de cada modelo en el conjunto de entrenamiento y en la tabla 47 los resultados obtenidos en el conjunto de validación.

Tabla 46

Comparativa de resultados dice loss, mean dice coefficient y mean IoU (entrenamiento)

Modelo	Año	Mean Dice Coefficient	Mean IoU	Dice Loss
U-Net	2015	0.8673	0.7833	0.1326
U-Net Residual Balanceada	2021	0.9521	0.9096	0.0478
U-Net Residual Desbalanceada	2021	0.9865	0.9734	0.0135
U-Net Doble	2021	0.9364	0.8827	0.0636

MU-Net	2023	0.9149	0.8477	0.0851
U-NetAtt	2024	0.8644	0.8109	0.1355

Nota. Elaboración propia

Los resultados obtenidos con el conjunto de entrenamiento muestran que el modelo U-Net Residual entrenado con el dataset desbalanceado es la variante con mejor desempeño, alcanzando un Mean Dice Coefficient de 0.9865, un Mean IoU de 0.9734, y un Dice Loss de 0.0135, lo que refleja una excelente precisión y una pérdida mínima durante el proceso de segmentación. Sin embargo, cabe destacar que este modelo se entrenó en un dataset desbalanceado, lo que puede favorecer el aprendizaje de las clases mayoritarias en detrimento de las minoritarias, afectando su generalización.

Por otro lado, la U-Net Residual entrenada con el dataset balanceado muestra resultados competitivos con un Mean Dice Coefficient de 0.9521, un Mean IoU de 0.9096, y un Dice Loss de 0.0478. Este desempeño, aunque ligeramente inferior al modelo desbalanceado, es más representativo debido al balance de las clases en el dataset, lo que sugiere una segmentación más equitativa y robusta entre las distintas categorías.

La U-Net Doble también exhibe un desempeño destacado, obteniendo un Mean Dice Coefficient de 0.9364 y un Mean IoU de 0.8827, posicionándose como una de las mejores variantes tras las U-Net Residual. Por su parte, el modelo MU-Net presentó métricas ligeramente inferiores, con un Mean Dice Coefficient de 0.9149 y un Mean IoU de 0.8477, pero aún se mantiene dentro de un rango aceptable para la tarea de segmentación.

En contraste, la variante U-Net con módulos de atención (U-NetAtt) tuvo el rendimiento más bajo del conjunto de modelos evaluados, alcanzando un Mean Dice Coefficient de 0.8644 y un Mean IoU de 0.8109, acompañado de un Dice Loss de 0.1355, lo que indica dificultades en la segmentación precisa.

Tabla 47

Comparativa de resultados dice loss, mean dice coefficient y mean IoU (conjunto validación)

Modelo	Año	Mean Dice Coefficient	Mean IoU	Dice Loss
U-Net	2015	0.7576	0.6542	0.2424
U-Net Residual Desbalanceada	2021	0.8235	0.7755	0.1765
U-Net Residual Balanceada*	2021	0.8763	0.8021	0.1236

U-Net Doble	2021	0.7529	0.6488	0.2470
MU-Net	2023	0.7764	0.6790	0.2235
U-NetAtt	2024	0.6647	0.6150	0.3352

Nota. Elaboración propia

*Resultados obtenidos en el conjunto de pruebas (10%)

En el conjunto de validación, los resultados destacan al modelo U-Net Residual Balanceada como el de mejor desempeño. Este modelo alcanzó un Mean Dice Coefficient de 0.8763, un Mean IoU de 0.8021, y un Dice Loss de 0.1236, lo que refleja una alta precisión en la segmentación. Cabe mencionar que estos resultados provienen del conjunto de prueba (10%), lo que valida su capacidad de generalización sobre datos no vistos previamente, además de evidenciar el impacto positivo del balanceo de las clases en el dataset.

En segundo lugar, el modelo U-Net Residual Desbalanceada también obtuvo un desempeño notable con un Mean Dice Coefficient de 0.8235, un Mean IoU de 0.7755, y un Dice Loss de 0.1765. Aunque ligeramente inferior al modelo balanceado, sigue mostrando una buena capacidad de segmentación. Sin embargo, su entrenamiento en un dataset desbalanceado podría haber favorecido las clases mayoritarias, lo que podría dificultar su desempeño en aplicaciones donde las clases minoritarias son igualmente relevantes.

Por su parte, el modelo MU-Net se posiciona como la tercera mejor variante, alcanzando un Mean Dice Coefficient de 0.7764, un Mean IoU de 0.6790, y un Dice Loss de 0.2235, lo que sugiere que, aunque es funcional, su precisión y pérdida son inferiores a las variantes residuales.

El modelo U-Net Doble presentó un desempeño inferior en comparación con los modelos previamente mencionados, con un Mean Dice Coefficient de 0.7529, un Mean IoU de 0.6488, y un Dice Loss de 0.2470, lo que indica dificultades en la segmentación precisa de las clases. Asimismo, el modelo U-Net original obtuvo métricas similares, con un Mean Dice Coefficient de 0.7576 y un Mean IoU de 0.6542, mostrando un rendimiento básico en la segmentación.

Finalmente, el modelo U-Net con módulos de atención (U-NetAtt) tuvo el desempeño más bajo del conjunto evaluado, con un Mean Dice Coefficient de 0.6647, un Mean IoU de 0.6150, y un Dice Loss de 0.3352, lo que indica que no logró segmentar las clases de manera precisa y uniforme.

En la tabla 48 se muestra una comparación de todos los resultados IoU por clase para cada modelo usando el conjunto de validación.

Tabla 48*Resultados mean IoU para cada clase para todos los modelos*

Modelo	IoU Background	IoU Lengua de vaca	IoU Diente de león	IoU Kikuyo	IoU Otras	IoU Papa	Mean IoU
U-Net	0.9693	0.6308	0.5826	0.6546	0.1832	0.9040	0.6541
U-Net Residual Desbalanceada	0.9676	0.7626	0.7404	0.7053	0.5564	0.9204	0.7755
U-Net Residual Balanceada*	0.9721	0.7984	0.7409	0.7730	0.5907	0.9376	0.8021
U-Net Doble	0.9659	0.6812	0.5648	0.6061	0.1782	0.8965	0.6488
MU-Net	0.9706	0.7463	0.5627	0.6763	0.2010	0.9174	0.6790
U-NetAtt	0.9600	0.7122	0.4811	0.5819	0.1274	0.8271	0.6150

Nota. Elaboración propia

*Resultados obtenidos en el conjunto de pruebas (10%)

En cuanto al análisis de las clases individuales, se resalta al modelo U-Net Residual Balanceada como la variante con mejor desempeño global, logrando un Mean IoU de 0.8021. Este modelo sobresale particularmente en las clases más desafiantes, como Kikuyo (0.7730) y Otras (0.5907), lo que demuestra que el balanceo de clases aplicado al dataset contribuyó significativamente a mejorar la segmentación de las clases minoritarias y con menor representación inicial.

En comparación, el modelo U-Net Residual Desbalanceada, aunque obtuvo buenos resultados con un Mean IoU de 0.7755, presentó un desempeño inferior en clases clave como Kikuyo (0.7053) y Otras (0.5564). Esto refuerza la hipótesis de que un dataset balanceado permite una segmentación más equitativa entre todas las clases, reduciendo el sesgo hacia las clases mayoritarias.

Por otro lado, modelos como MU-Net (Mean IoU de 0.6790) y U-Net Doble (Mean IoU de 0.6488) mostraron resultados menos consistentes, con bajos desempeños en las clases Otras (0.2010 y 0.1782, respectivamente), lo que evidencia su dificultad para manejar la segmentación en escenarios desbalanceados. Finalmente, el modelo U-Net con módulos de atención (U-NetAtt) obtuvo el rendimiento más bajo con un Mean IoU de 0.6150, afectado principalmente por su desempeño en las clases minoritarias, como Otras (0.1274).

3.1.2. Análisis de resultados y elección del mejor modelo

La elección del mejor modelo se fundamenta en un análisis detallado de las métricas obtenidas durante el entrenamiento, validación y rendimiento por clase. En esta investigación, la U-Net Residual Balanceada ha demostrado ser la variante con mejor desempeño global en múltiples aspectos clave. Durante el entrenamiento, este modelo alcanzó un Mean Dice Coefficient de 0.9521 y un Mean IoU de 0.9096, junto con un Dice Loss de 0.0478, reflejando una excelente capacidad para segmentar las imágenes del conjunto de entrenamiento. En el conjunto de pruebas, mantuvo un rendimiento destacado con un Mean Dice Coefficient de 0.8763, un Mean IoU de 0.8021, y un Dice Loss de 0.1236, lo que evidencia su capacidad de generalización a datos no vistos.

El análisis por clase refuerza esta elección. La U-Net Residual Balanceada obtuvo resultados sobresalientes en clases clave como Lengua de vaca, con un IoU de 0.7984, y Papa, con un IoU de 0.9376, superando a las demás variantes evaluadas. Este desempeño se extiende a clases desafiantes como Kikuyo (IoU de 0.7730) y Otras (IoU de 0.5907), lo que destaca la importancia del balanceo de clases aplicado al dataset, especialmente para mejorar la segmentación en categorías con menor representación.

En contraste, la U-Net Residual Desbalanceada, aunque mostró un rendimiento alto con un Mean IoU de 0.7755 y métricas similares en clases como Lengua de vaca (IoU de 0.7626), presentó un menor desempeño en clases como Kikuyo (IoU de 0.7053) y Otras (IoU de 0.5564), evidenciando un sesgo hacia las clases mayoritarias. Por otro lado, modelos como la U-Net Doble y la MU-Net, aunque lograron métricas competitivas en el conjunto de entrenamiento, con un Mean IoU de 0.8827 y 0.8477, respectivamente, tuvieron un rendimiento menos consistente en validación, con una marcada caída en clases minoritarias. Finalmente, la U-Net con módulos de atención (U-NetAtt) mostró el desempeño más bajo, con un Mean IoU de 0.6150, especialmente afectado por su rendimiento en clases como Otras (IoU de 0.1274).

Por lo tanto, la U-Net Residual Balanceada es elegida como el mejor modelo debido a su capacidad para mantener un alto rendimiento tanto en entrenamiento como en validación, así como su excelente desempeño en la segmentación por clases clave del conjunto de datos.

3.2. Validación según la Norma ISO 25023

La segunda validación se realizará de acuerdo con la norma ISO 25023, enfocándose en la característica de comportamiento del tiempo. Específicamente, se utilizará la métrica de tiempo de respuesta para evaluar el tiempo necesario para completar una tarea específica utilizando la solución tecnológica propuesta en este trabajo de investigación. La ISO 25023 establece directrices para medir la eficiencia temporal de sistemas y aplicaciones, y en este contexto, el objetivo es estimar el tiempo requerido para procesar una imagen y realizar la

segmentación de malezas. En la figura 63 se detalla la forma de aplicar dicha validación.

Figura 63

Validación sobre la subcaracterística Comportamiento del tiempo de la ISO 25023

Métricas para la característica de calidad Eficiencia en el desempeño								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Comportamiento del tiempo	Tiempo de respuesta	Interna/Externa	¿Cuál es el tiempo estimado para completar una tarea?	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	$X = B - A$ A= Tiempo de envío de petición B = Tiempo en recibir la primera respuesta	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$.	X= Tiempo/Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Tiempo de espera	Interna/Externa	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	$X = B - A$ A= Tiempo cuando se inicia un trabajo B = Tiempo en completar el trabajo	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$.	X= Tiempo/Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Nota. Extraído de: (Larios Mar, 2021)

Esta validación permitirá determinar si la solución tecnológica no solo es precisa, sino también eficiente en términos de tiempo, lo cual es crucial para su implementación práctica en entornos agrícolas.

Para llevar a cabo esta validación, se definió un entorno de pruebas controlado en el que se procesaron múltiples imágenes representativas de cultivos de papa. Además, se utilizó la librería System.Diagnostics, que proporciona clases que permiten interactuar con los procesos del sistema, registros de eventos y contadores de rendimiento. En este contexto, se evaluará el tiempo de respuesta de 10 solicitudes HTTP y el tamaño de los datos enviados y recibidos. Los resultados se detallan en la tabla 49 y en la figura 64.

Tabla 49

Resultados de respuestas HTTP

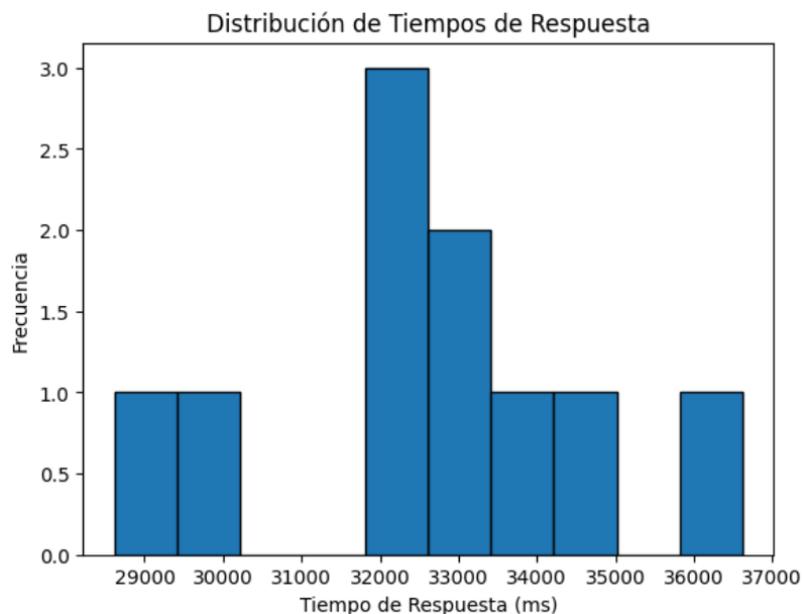
Número solicitud	Tiempo de respuesta (ms)	Tamaño de datos solicitud (bytes)	Tamaño de datos respuesta (bytes)
1	32720	555866	22788
2	32631	654440	36479
3	31891	845689	12363
4	28616	559036	21134
5	31974	777693	52496

6	33447	549965	42199
7	29442	506060	24415
8	31949	496351	45019
9	34995	472129	34331
10	36618	499668	25686

Nota. Elaboración propia

Figura 64

Distribución de tiempos de respuesta



Nota. Elaboración propia

En base a estos resultados se calculó el tiempo de respuesta promedio y su desviación estándar utilizando las siguientes fórmulas:

$$\text{Promedio} = \frac{\Sigma \text{ Tiempos de Respuesta}}{N} \qquad \text{Desviación Estandar} = \sqrt{\frac{\Sigma (x_i - \mu)^2}{N}}$$

Donde:

- N es el número total de solicitudes.
- x_i es cada tiempo de respuesta.
- μ es el tiempo de respuesta promedio.

El tiempo de respuesta promedio y la desviación estándar son métricas fundamentales para evaluar el desempeño temporal de un sistema. El primero indica cuánto tiempo, en promedio, tarda el sistema en procesar una solicitud. En este caso, el tiempo de respuesta promedio fue de 32428.3 milisegundos. Este valor es útil para entender la eficiencia general del sistema y proporciona una medida estándar para comparar con otros sistemas o con futuros desarrollos del mismo tema.

La desviación estándar, por otro lado, mide la dispersión de los tiempos de respuesta respecto al promedio. Una desviación estándar baja indica que los tiempos de respuesta son consistentes y están cerca del promedio, mientras que una desviación estándar alta indica una mayor variabilidad en los tiempos de respuesta. En este caso, su valor fue de 2224.45 milisegundos, lo que sugiere que hay cierta variabilidad en los tiempos de respuesta, pero no es excesivamente alta.

Estos resultados son necesarios para entender la confiabilidad y consistencia del sistema en situaciones prácticas. Un tiempo de respuesta promedio bajo con una desviación estándar baja es ideal, ya que indica que el sistema no solo es rápido, sino también consistente en su desempeño. Sin embargo, si el tiempo de respuesta promedio es alto o la desviación estándar es alta, se deben identificar y abordar los factores causantes de esta variabilidad para mejorar el rendimiento del sistema. En la figura 65 se visualiza los tiempos de respuesta para cada solicitud individual, mientras que en la figura 66 se muestra el promedio y la variabilidad de los tiempos de respuesta.

Figura 65

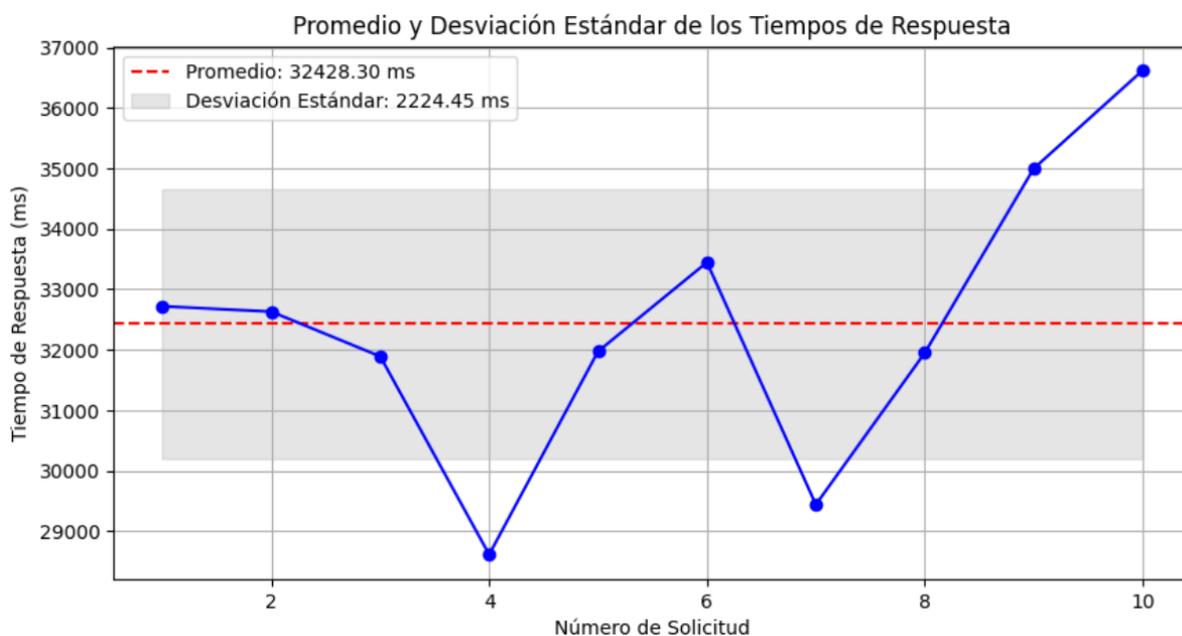
Tiempos de respuesta por solicitud



Nota. Elaboración propia

Figura 66

Promedio y desviación estándar de los tiempos de respuesta



Nota. Elaboración propia

3.3. Comparación de tiempos entre el método manual y el propuesto

La problemática abordada en este trabajo de investigación consiste en reducir el tiempo dedicado a la identificación y cuantificación de malezas en cultivos de papa, que tradicionalmente se lleva a cabo mediante la observación visual de los agricultores. Este método no solo es subjetivo, sino que también es muy dependiente de la experiencia del agricultor y puede variar en función del terreno, las condiciones climáticas y la extensión del área evaluada.

Para determinar si el sistema automatizado desarrollado en este trabajo mejora este proceso, se ha decidido seleccionar 3 parcelas de cultivos y comparar el tiempo empleado en la identificación y cuantificación de malezas de forma manual versus el tiempo tomado por el sistema basado en visión por computadora. La comparación será fundamentada mediante la prueba estadística t de Student (Student, 1908), que permite evaluar si existe una diferencia significativa entre las medias de ambos métodos, proporcionando un análisis estadístico objetivo de los resultados obtenidos.

Según Lind et al. (2012) la ecuación para calcular el valor t es:

$$t = \frac{d}{S_d/\sqrt{n}}$$

Donde:

- d es la media de la diferencia.
- S_d es la desviación estándar de la diferencia.
- n es el número de muestras.

Además, se define la hipótesis nula y alterna:

$$H_0: d = 0$$

$$H_1: d \neq 0$$

Donde:

- H_0 es la hipótesis nula, plantea que no hay una diferencia significativa en la media de la diferencia.
- H_1 es la hipótesis alterna, plantea que si existe diferencia significativa.

En la tabla 50 se describen los tiempos para cada terreno seleccionado al realizar la tarea de identificación y cuantificación de malezas en cultivos de papa de manera tradicional.

Tabla 50

Tiempo empleado para identificar y cuantificar malezas en los 3 terrenos seleccionados de forma tradicional

Terreno	Tamaño (m ²)	Tiempo (minutos)
Terreno 1	730	60
Terreno 2	800	66
Terreno 3	650	54

Nota. Elaboración propia

En la tabla 51 se describen los tiempos para cada terreno seleccionado al realizar la tarea de identificación y cuantificación de malezas en cultivos de papa usando la solución tecnológica propuesta en este trabajo de investigación. Este tiempo abarca desde que se vuela el dron para capturar las fotos que posteriormente serán procesadas en el sistema.

Tabla 51

Tiempo empleado para identificar y cuantificar malezas en los 3 terrenos seleccionados usando el aplicativo móvil

Terreno	Tamaño (m ²)	Tiempo (minutos)
---------	--------------------------	------------------

Terreno 1	730	51
Terreno 2	800	55
Terreno 3	650	44

Nota. Elaboración propia

Para realizar la prueba t Student automáticamente se utilizó la librería scipy de Python. En la tabla 52 se exponen los resultados obtenidos, en donde TT es el conjunto de tiempos obtenidos de manera tradicional y TA es el conjunto de tiempos obtenidos por el aplicativo móvil.

Tabla 52

Prueba t Student a los conjuntos de tiempo TT y TA

Medida	TT	TA
Media	60	50
Varianza	36	31
Observaciones	3	3
Coefficiente de correlación de Pearson	0.9878	
Diferencia hipotética de las medias	0	
Grados de libertad	2	
Estadístico t	17.3205	
P(T<=t) una cola	0.0016	
Valor crítico de t (una cola)	2.9199	
P-valor	0.0033	
Valor crítico de t (dos colas)	4.3026	

Nota. Elaboración propia

El análisis de la prueba t de Student indica una diferencia significativa en el tiempo empleado para identificar y cuantificar malezas entre el método tradicional y el uso del aplicativo móvil. La media de los tiempos obtenidos para el método tradicional fue de 60 minutos, mientras que el uso del aplicativo móvil redujo este tiempo a una media de 50

minutos. Las varianzas de ambos métodos, 36 para el método tradicional y 31 para el aplicativo móvil, muestran una ligera variación en la dispersión de los datos.

En cuanto a la prueba t, el estadístico t obtenido fue de 17.32, con un p-valor de 0.0033 (dos colas), significativamente menor que el umbral de 0.05. Esto permite rechazar la hipótesis nula (H_0), que planteaba que no habría diferencia significativa entre los métodos, en favor de la hipótesis alterna (H_1), la cual establece que sí existe una diferencia.

Después del análisis se puede concluir que la prueba t confirma que el aplicativo móvil reduce considerablemente el tiempo necesario para identificar y cuantificar malezas en los terrenos evaluados. Este resultado es estadísticamente significativo y valida la eficiencia del aplicativo móvil en comparación con el método tradicional.

Discusión

Dentro del trabajo de investigación se han abarcado múltiples versiones de la red neuronal convolucional U-Net, haciendo énfasis en las adaptaciones realizadas para poder emplearlas en la tarea de segmentación semántica de malezas en cultivos de maíz. En el desarrollo e implementación de cada arquitectura se hicieron varios experimentos cambiando los valores para los hiperparámetros de entrenamiento, con el fin de encontrar el mejor modelo para esta tarea. Fueron 5 arquitecturas que se dieron a conocer a lo largo de este trabajo, de las cuales 4 fueron inspiradas de diferentes autores con la particularidad de que su arquitectura fue adaptada y en parte cambiada para poder utilizarla con el enfoque que se necesitaba. La última arquitectura fue propuesta a una variante del modelo original de U-Net recogiendo características usadas dentro del mundo de las redes neuronales convolucionales.

A partir de los resultados obtenidos se concluyó que la arquitectura U-Net Residual Balanceada fue la que presentó mejores resultados en comparación de las demás y fue la que se implementó dentro de la aplicación móvil.

A continuación en la tabla 53 se muestra los resultados de los trabajos relacionados con la identificación de malezas usando arquitecturas semejantes a U-Net. Cabe recalcar que en la mayoría investigaciones se enfocan principalmente en 2 clases, cultivo y maleza. Sin embargo, hay tres propuestas en la que se enfocan en más clases, sus resultados se detallan en la tabla 54 y 55.

Tabla 53

Resultados de trabajos relacionados enfocados en 2 clases

Autores	Modelo	Cultivo	Número de malezas	Mean IoU %	Precision %
(Cui J, Tan F, Bai N and Fu Y, 2024)	Pspnet	Repollo chino	1 (no específica)	71.64	83.17
(Cui J, Tan F, Bai N and Fu Y, 2024)	Deeplab V3	Repollo chino	1 (no específica)	76.50	75.61
(Cui J, Tan F, Bai N and Fu Y, 2024)	RDS_U-Net	Repollo chino	1 (no específica)	82.36	91.36
(Shanwen Zhang, Chuanlei Zhang, 2023)	MFCN	1 (No específica)	1 (Plantas enfermas)	83.66	89.12
(Shanwen Zhang, Chuanlei Zhang,	Res_U-Net	1 (No específica)	1 (Plantas enfermas)	89.32	92.37

2023)					
(Shanwen Zhang, Chuanlei Zhang, 2023)	MultiR esU- Net	1 (No especifica)	1 (Plantas enfermas)	90.64	93.46
(Shanwen Zhang, Chuanlei Zhang, 2023)	MU- Net	1 (No especifica)	1 (Plantas enfermas)	94.07	95.13
(Kong, X., Liu, T., Chen, X., Jin, X., Li, A., & Yu, J., 2024)	ECSN et	Maíz	1 (no especifica)	90.9	90.2
(Zuo, Y., & Li, W., 2024)	MN_P PM_M D_unet	Maíz	4 (Bluegrass, Chenopodium album, Cirsium setosum, Sedge)	87.9	99.3
(Jiang, H., Chen, Q., Wang, R., Du, J., & Chen, T., 2024)	SWFor mer	9 (8 clases de cultivo del cropandweed dataset y remolacha azucarera del dataset SB20)	1 (Agrupa todas las clases de malezas no especificadas en una)	76.54	83.95

Nota. Elaboración propia

Tabla 54

Resultados de trabajos relacionados enfocados en 3 clases

Autores	Modelo	Cultiv o	Número de malezas	Mean IoU Backgro und %	Mean IoU Maleza %	Mean IoU Cultivo %	Mean IoU %
----------------	---------------	---------------------	----------------------------------	---	--------------------------------------	---------------------------------------	-----------------------

(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	PSPNet	Maíz	1 (no específico a)	98.31	38.67	87.81	74.93
(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	Deeplab V3+	Maíz	1 (no específico a)	99.02	63.43	92.71	85.06
(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	MSECA- Unet	Maíz	1 (no específico a)	99.24	73.62	94.02	88.96
(Liao, J., Chen, M., Zhang, K., Zhou, H., Zou, Y., Xiong, W., Zhang, S., Kuang, F., & Zhu, D., 2024)	SC-Net	Arroz	2 (gramineae weeds, broadleaf weeds)	97.01	81.92	83.51	87.48

Nota. Elaboración propia

Tabla 55

Resultados de trabajos relacionados enfocados en las mismas clases utilizadas

Autor	Modelo	Cultivo	IoU Background %	IoU Lengua de vaca %	IoU Diente de león %	IoU Kikuyo %	IoU Otras %	IoU Papa %	Mean IoU %
(Vinuesa K.,	ResNeXt 50	Papa	95.6	82.6	58.3	70.3	43.4	90.8	73.5

2024) modificad

a

Nota. Elaboración propia

Se puede observar que 2 trabajos de investigación utilizaron el enfoque de 2 clases, siendo cultivo y maleza; 1 se centra en 3 clases, Background, maleza, cultivo y el último en las clases utilizadas en este trabajo de investigación siendo Background, lengua de vaca, diente de león, kikuyo, otras y papa. Para poder comparar el trabajo realizado con las demás propuestas se decidió tomar el mean IoU y precisión general del modelo en el primer tipo de investigación, el enfoque para el segundo tipo fue hacer un promedio de las malezas en las que se centra esta investigación y finalmente el último se realizara una comparación completa. Las comparativas se muestran en la tabla 56, 57 y 58 para cada tipo de investigación respectivamente.

Tabla 56

Comparativa con trabajos relacionados enfocados en 2 clases (cultivo, maleza)

Autores	Modelo	Cultivo	Número de malezas	Mean IoU %	Precisión %
(Cui J, Tan F, Bai N and Fu Y, 2024)	Pspnet	Repollo chino	1 (no específica)	71.64	83.17
(Cui J, Tan F, Bai N and Fu Y, 2024)	Deeplab V3	Repollo chino	1 (no específica)	76.50	75.61
(Cui J, Tan F, Bai N and Fu Y, 2024)	RDS_U-Net	Repollo chino	1 (no específica)	82.36	91.36
(Shanwen Zhang, Chuanlei Zhang, 2023)	MFCN	1 (No específica)	1 (Plantas enfermas)	83.66	89.12
(Shanwen Zhang, Chuanlei Zhang, 2023)	Res_U-Net	1 (No específica)	1 (Plantas enfermas)	89.32	92.37
(Shanwen Zhang, Chuanlei Zhang, 2023)	MultiResU-Net	1 (No específica)	1 (Plantas enfermas)	90.64	93.46
(Shanwen Zhang, Chuanlei Zhang, 2023)	MU-Net	1 (No específica)	1 (Plantas enfermas)	94.07	95.13

2023)					
(Kong, X., Liu, T., Chen, X., Jin, X., Li, A., & Yu, J., 2024)	ECSN et	Maíz	1 (no específica)	90.9	90.2
(Zuo, Y., & Li, W., 2024)	MN_P PM_M D_unet	Maíz	4 (Bluegrass, Chenopodium album, Cirsium setosum, Sedge)	87.9	99.3
(Jiang, H., Chen, Q., Wang, R., Du, J., & Chen, T., 2024)	SWFor mer	9 (8 clases de cultivo del cropandweed dataset y remolacha azucarera del dataset SB20)	1 (Agrupa todas las clases de malezas no especificadas en una)	76.54	83.95
Jorge Pazos (autor del trabajo de titulación)	U-Net Residu al adapta da	Papa	3 (lengua de vaca, kikuyo y diente de león)	85.41	96.47 (en el conjunto de pruebas)

Nota. Elaboración propia

La arquitectura adaptada de U-Net Residual muestra un Mean IoU general del 85.41%, ubicándose por debajo de los modelos MU-Net, MultiResU-Net, Res_U-Net, MFCN y RDS_U-Net, pero superando a los modelos DeepLab V3 y Pspnet. Esto permite posicionar a la U-Net Residual adaptada en un rango intermedio dentro de la comparativa con los trabajos relacionados enfocados en la segmentación de dos clases (cultivo y maleza).

Al analizar los resultados de este estudio y compararlos con investigaciones previas, la U-Net Residual adaptada demuestra ser una solución competitiva en tareas de segmentación de malezas en cultivos agrícolas. Este modelo fue específicamente diseñado para segmentar tres tipos de malezas en cultivos de papa: lengua de vaca, kikuyo y diente de león, lo que añade complejidad a la tarea de segmentación en comparación con los estudios revisados, los cuales se enfocaron en una sola clase de malezas. Con un Mean IoU del 85.41% y una precisión del 96.47% en el conjunto de pruebas, el modelo destaca por su

capacidad para abordar un escenario con mayor variedad de clases, logrando una segmentación precisa incluso en un contexto más desafiante.

En trabajos recientes, como los realizados por Cui et al. (2024) y Zhang et al. (2023), se observa el uso de arquitecturas avanzadas de segmentación, tales como Pspnet, Deeplab V3, RDS_U-Net, MFCN, Res_U-Net, y variantes optimizadas como MultiResU-Net y MU-Net, aplicadas principalmente en cultivos menos diversos, como el repollo chino y plantas enfermas. Dichos estudios reportan un Mean IoU superior, alcanzando hasta un 94.07% en el caso del MU-Net, lo que podría atribuirse a la simplicidad de trabajar con una sola clase de maleza. No obstante, la U-Net Residual adaptada en este trabajo supera a muchos de estos modelos en términos de precisión, destacándose con un 96.47%, superior a la precisión reportada por modelos como el MultiResU-Net (93.46%) y el MU-Net (95.13%).

Asimismo, se incorporaron dos trabajos recientes del 2024. El modelo ECSNet, desarrollado por Kong et al., logra un Mean IoU de 90.9% y una precisión de 90.2% en cultivos de maíz con una clase de maleza no especificada, destacándose por su equilibrio entre precisión y desempeño computacional. Por otro lado, el MN_PPM_MD_unet, presentado por Zuo y Li, reporta un Mean IoU de 87.9% y una precisión sobresaliente de 99.3% al segmentar cuatro clases específicas de malezas en maíz, lo que evidencia la capacidad del modelo para abordar un escenario más diversificado. Estos resultados resaltan que los avances en técnicas de segmentación pueden ofrecer mejoras sustanciales en precisión y robustez, aunque las diferencias en cultivos y clases segmentadas dificultan comparaciones directas.

Adicionalmente, el modelo SWFormer, propuesto por Jiang et al. (2024), introduce un enfoque híbrido CNN-Transformer diseñado para la segmentación de cultivos y malezas en datasets públicos como cropandweed y SB20. Este modelo combina las capacidades de las CNN para capturar dependencias locales y de los transformers para detectar relaciones globales, logrando un Mean IoU del 76.54% y una precisión del 83.95%. Aunque el SWFormer aborda un mayor rango de clases de cultivos, su rendimiento en términos de IoU es inferior al de la U-Net Residual adaptada, lo que destaca la efectividad de esta última en el contexto específico de cultivos de papa con múltiples clases de malezas.

Por lo tanto, aunque la media de IoU del modelo aquí desarrollado es menor en comparación con estudios que solo abordan una clase de maleza, la alta precisión refleja su capacidad para identificar con exactitud las diferentes clases de malezas en cultivos de papa, un contexto considerablemente más complejo. Este análisis reafirma la efectividad de la U-Net Residual adaptada para entornos agrícolas con múltiples clases, donde se prioriza no solo la exactitud en la segmentación, sino también la robustez del modelo en escenarios diversos.

Tabla 57*Comparativa con trabajos relacionados enfocados en 3 clases (Background, cultivo, maleza)*

Autores	Modelo	Cultivo	Número de malezas	Mean IoU Background %	Mean IoU Maleza %	Mean IoU Cultivo %	Mean IoU %
(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	PSPNet	Maíz	1 (no específico a)	98.31	38.67	87.81	74.93
(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	DeepLab V3+	Maíz	1 (no específico a)	99.02	63.43	92.71	85.06
(Ma Z., Wang G., Yao J., Huang D., Tan H., Jia H., Zou Z., 2023)	MSECA-UNET	Maíz	1 (no específico a)	99.24	73.62	94.02	88.96
(Liao, J., Chen, M., Zhang, K., Zhou, H., Zou, Y., Xiong, W., Zhang, S., Kuang, F., & Zhu, D., 2024)	SC-Net	Arroz	2 (gramineae weeds, broadleaf weeds)	97.01	81.92	83.51	87.48

Jorge Pazos (autor del trabajo de titulación)	U-Net Residual adaptada	Papa	3 (lengua de vaca, kikuyo y diente de león)	97.21	77.07 (promedi o de lengua de vaca, kikuyo y diente de león)	93.76	89.34 (prome dio de backgr ound, maleza s y cultivo)
--	-------------------------------	------	---	-------	---	-------	---

Nota. Elaboración propia

La comparación de la U-Net Residual adaptada con trabajos recientes que segmentan tres clases muestra que el modelo desarrollado en este trabajo es altamente efectivo en la identificación de malezas en un contexto complejo como el cultivo de papa. En términos de Mean IoU para la clase de malezas, el modelo alcanza un 77.07%, superando el desempeño de arquitecturas avanzadas como la MSECA-Unet, que reporta un 73.62% en un cultivo de maíz (Ma et al., 2023). Esto evidencia la capacidad de la U-Net Residual adaptada para segmentar con precisión múltiples tipos de malezas, a pesar de la mayor diversidad de clases incluidas en este trabajo.

Para la clase background, la U-Net Residual adaptada logra un 97.21% de IoU, manteniéndose competitiva frente a modelos como la MSECA-Unet (99.24%) y la DeepLab V3+ (99.02%). Aunque estos valores son ligeramente superiores, es importante considerar que los estudios comparados trabajan con un escenario menos desafiante, al abordar únicamente una clase de maleza no especificada, lo que simplifica el proceso de segmentación. Por otro lado, la clase cultivo alcanza un 93.76% de IoU, un resultado que refleja la capacidad del modelo para segmentar eficazmente esta categoría, siendo comparable con los valores reportados por otras arquitecturas avanzadas.

El modelo SC-Net (Liao et al., 2024), aplicado al arroz, muestra un alto desempeño en la clase de maleza, alcanzando un 81.92% de IoU, lo que lo coloca por encima de la U-Net Residual adaptada (77.07%). Sin embargo, es relevante señalar que SC-Net trabaja con dos clases de maleza específicas (gramíneas y malezas de hoja ancha), lo que puede facilitar la segmentación en comparación con el enfoque más general de la U-Net Residual adaptada. Además, la clase background en SC-Net alcanza un 97.01% de IoU, muy similar al 97.21% obtenido por la U-Net Residual adaptada, lo que sugiere que ambos modelos manejan bien esta categoría. La clase cultivo en SC-Net alcanza un 83.51%, un poco por debajo del 93.76% alcanzado por la U-Net Residual adaptada, lo que demuestra la efectividad superior de esta última en el cultivo de papa.

Finalmente, la Mean IoU global de la U-Net Residual adaptada alcanza un 89.34%, superando a modelos como la PSPNet (74.93%) y situándose ligeramente por encima de la MSECA-Unet (88.96%). Al comparar con el SC-Net, la U-Net Residual adaptada muestra una ligera ventaja en términos de segmentación global, destacando su robustez y efectividad en un escenario con mayor diversidad de clases. Estos resultados posicionan a la U-Net Residual adaptada como una herramienta sólida para la segmentación de múltiples clases de malezas en cultivos agrícolas, demostrando un balance adecuado entre precisión y generalización en contextos complejos.

Tabla 58

Comparativa con trabajos relacionados enfocados en las mismas clases utilizadas

Autor	Modelo	Cultivo	IoU Background %	IoU	IoU	IoU Kikuyo %	IoU Otras %	IoU Papa %	Mean IoU %
				Lengua de vaca %	Diente de león %				
(Vinueza K, 2024)	ResNet50 modificada	Papa	95.6	82.6	58.3	70.3	43.4	90.8	73.5
Jorge Pazos (autor del trabajo de titulación)	U-Net Residual Balanceada	Papa	96.76	76.26	74.04	70.53	55.64	92.04	77.55
Jorge Pazos (autor del trabajo de titulación)	U-Net Residual Balanceada	Papa	97.21	79.84	74.09	77.30	59.07	93.76	80.21

Nota. Elaboración propia

La comparación con el modelo ResNet50 modificado de Vinueza K. (2024), desarrollado para un contexto similar y en el mismo cultivo de papa, permite una evaluación precisa del avance logrado en este trabajo. La arquitectura U-Net Residual Balanceada superó

el desempeño del modelo de Vinueza en términos de Mean IoU, alcanzando un 80.21% frente al 73.5% de la ResNeXt50 modificada. Esta mejora se atribuye, en gran parte, al enfoque en el balance de clases, ya que el dataset utilizado en el trabajo de Vinueza presentaba un notable desbalance, lo cual afectó el reconocimiento de ciertas clases, como la clase de Otras malezas, que obtuvo un IoU de apenas 43.4% en su modelo.

Por otro lado, la U-Net Residual Desbalanceada, aunque trabajó con un dataset con clases desigualmente representadas, también logró superar el modelo ResNeXt50 de Vinueza en términos de Mean IoU, obteniendo un 77.55%. Este modelo destacó especialmente en la clase de Diente de león, con un IoU de 74.04%, y en la clase de Papa, con un IoU de 92.04%, superando los valores de 58.3% y 90.8% reportados por el modelo de Vinueza. Sin embargo, el desbalance en las clases afectó su rendimiento en la clase de Lengua de vaca, donde obtuvo un IoU de 76.26% frente al 82.6% del modelo ResNeXt50.

El modelo U-Net Residual Balanceada, en cambio, muestra una mejora considerable en casi todas las clases. En la clase Diente de león, alcanzó un IoU de 74.09%, superando tanto al modelo de Vinueza (58.3%) como a la versión desbalanceada (74.04%). En la clase de Otras malezas, logró un IoU de 59.07%, significativamente superior al 43.4% del modelo ResNeXt50 y al 55.64% de la versión desbalanceada. Además, este modelo también sobresale en las clases de Background y Papa, con un IoU de 97.21% y 93.76%, respectivamente, mostrando un rendimiento más uniforme y robusto en comparación con los otros modelos.

Estos resultados evidencian que el balance del dataset juega un papel crucial en la mejora del rendimiento del modelo, favoreciendo su capacidad para generalizar y segmentar múltiples clases de malezas de manera precisa. Si bien la U-Net Residual Desbalanceada también obtuvo resultados competitivos, es evidente que la versión balanceada proporciona una segmentación más uniforme y efectiva frente al trabajo relacionado y las variaciones de la propia U-Net Residual.

Esta investigación enfrentó varias limitaciones que afectaron tanto el desarrollo como los resultados finales. En primer lugar, el dataset, aunque balanceado para mejorar la precisión del modelo en distintas clases de malezas, estuvo limitado en su diversidad espacial y temporal. Las imágenes fueron capturadas bajo condiciones específicas de luz y clima, lo que puede afectar el desempeño del modelo en situaciones climáticas diferentes o en otros ciclos de cultivo. Adicionalmente, los recursos computacionales representaron una restricción, ya que la necesidad de procesar y entrenar modelos complejos en imágenes de alta resolución incrementa significativamente el costo y tiempo de procesamiento. Aunque el uso de servicios de computación en la nube, como Google Colab Pro, permitió entrenar el modelo de manera eficiente, la dependencia de estos servicios introduce una variable de acceso y disponibilidad económica que puede limitar la replicabilidad en contextos de menor capacidad de recursos.

Para avanzar en esta línea de investigación, sería útil extender la recolección de datos

a diferentes ciclos de cultivo y condiciones ambientales, aumentando así la capacidad de generalización del modelo. Además, explorar el uso de modelos de redes neuronales de mayor capacidad, como las arquitecturas de redes transformer, puede ayudar a mejorar la precisión en la segmentación y a reducir posibles sesgos introducidos por el entorno específico de entrenamiento. Asimismo, integrar métodos de preprocesamiento y técnicas de data augmentation más avanzadas puede proporcionar al modelo una mayor robustez. También sería valioso desarrollar un sistema de retroalimentación continua en campo que permita validar el modelo en situaciones reales, ajustando así los parámetros del sistema y obteniendo datos adicionales que alimenten y mejoren el modelo en tiempo real.

Estas recomendaciones no solo mejorarían la aplicabilidad del modelo en diversos contextos, sino que también impulsarían el desarrollo de una herramienta adaptable a las demandas cambiantes en la gestión de malezas, contribuyendo a prácticas agrícolas más eficaces y sostenibles.

Conclusiones

- La revisión bibliográfica ha demostrado que las redes neuronales convolucionales son herramientas altamente efectivas para la segmentación de imágenes en el ámbito agrícola. En particular, la arquitectura U-Net, inicialmente diseñada para tareas de segmentación en imágenes médicas, ha demostrado una notable capacidad de adaptación en campos como la agricultura. Esta flexibilidad ha permitido aplicar la U-Net para abordar problemáticas relacionadas con la identificación y cuantificación de malezas en cultivos. Además, con el constante desarrollo tecnológico las tareas que en un inicio carecían de métodos automatizados o que simplemente se basaban en procesos empíricos como costumbres y enseñanzas ancestrales se han visto beneficiadas. La utilización de drones para la captura de imágenes de terrenos de cultivos ofrece una ventaja significativa en términos de cobertura y resolución temporal y espacial, crucial para la cuantificación precisa de malezas. El dataset balanceado generado a partir de estas imágenes fue clave para mejorar la calidad del entrenamiento y garantizar la robustez de las predicciones, evidenciando su impacto positivo en la tarea de segmentación en comparación del dataset original que presentaba desbalance en sus clases.
- La implementación del modelo U-Net con todas sus versiones expuestas y desarrolladas en este trabajo de investigación han demostrado ser efectivas en la segmentación de malezas en imágenes capturadas por drones. Estas variantes incluyeron arquitecturas como la U-Net básica, U-Net Residual, U-Net Doble, MU-Net, U-Net con módulos de atención (U-NetAtt), y una U-Net Residual balanceada. El proceso de entrenamiento, estructurado y documentado, ha permitido obtener resultados satisfactorios en todas las versiones del modelo. Sin embargo, la arquitectura que ha demostrado un mejor desempeño ha sido la U-Net Residual con el dataset balanceado, mostrando una mejora notable en comparación con las demás variaciones evaluadas.
- El despliegue del mejor modelo mediante una API REST desarrollada en Flask y dockerizada asegura una fácil implementación y uso en distintos entornos, promoviendo la adopción práctica de la solución tecnológica, en este caso siendo un aplicativo móvil la elección para interactuar con el usuario final brindándole un acceso rápido al sistema desarrollado. La implementación práctica de la tecnología propuesta tiene el potencial de mejorar significativamente la eficiencia y efectividad del manejo de malezas, contribuyendo a prácticas agrícolas más sostenibles y productivas.

- Las validaciones realizadas a través de métricas como el mean dice coefficient, el dice loss y el mean IoU han confirmado que los modelos entrenados, en particular la U-Net Residual, ofrecen una alta precisión en la segmentación de malezas. Esto evidencia que la tecnología desarrollada es una herramienta útil para la cuantificación automática de malezas en los cultivos estudiados, lo cual contribuye a una gestión más eficiente y precisa en el ámbito agrícola.

Recomendaciones

- Se recomienda mantener una revisión continua de la literatura científica y técnica para estar al tanto de los avances más recientes en tecnología de redes neuronales y procesamiento de imágenes aplicadas a la agricultura. Es fundamental ampliar el uso de drones para capturar imágenes bajo diversas condiciones climáticas y de iluminación, y en distintos tipos de cultivos, con el objetivo de aumentar la robustez y aplicabilidad de los resultados obtenidos.
- Continuar refinando y optimizando la arquitectura original de U-Net, mediante ajustes de hiperparámetros y pruebas con nuevos conjuntos de datos, con el fin de mejorar su precisión y eficacia en tareas de segmentación semántica. Al tratarse de redes neuronales convolucionales complejas, se recomienda entrenarlas en entornos virtuales que ofrezcan servicios de cómputo de alto rendimiento, como Google Colab PRO, utilizado en este trabajo de investigación. Estas plataformas proporcionan los recursos computacionales necesarios para entrenamientos intensivos, evitando los altos costos de adquisición de hardware especializado y garantizando un entorno adecuado para tareas de procesamiento de datos de gran envergadura que no pueden ser ejecutadas en ordenadores comunes
- Se recomienda implementar un sistema de monitoreo y actualización continua para la API REST y el entorno dockerizado, asegurando que el sistema mantenga un desempeño óptimo y esté preparado para adaptarse a nuevos requerimientos y mejoras. Esto permitirá una mayor flexibilidad y escalabilidad del sistema, facilitando su integración en diversos contextos operativos.
- Se sugiere realizar validaciones periódicas y ampliadas del modelo con nuevos conjuntos de datos, bajo diferentes condiciones y escenarios de campo, para asegurar y mejorar la precisión del sistema. Además, sería beneficioso implementar evaluaciones prácticas en situaciones reales para medir el impacto del sistema en la eficiencia del manejo de malezas, ajustando las configuraciones del modelo en función de los resultados obtenidos. Este enfoque asegurará la evolución continua del sistema y su adaptabilidad a las necesidades del sector agrícola.

Referencias

- FAO. (2007). Recomendaciones para el manejo de malezas [Documento en línea]. Recuperado de <https://www.fao.org/3/a0884s/a0884s.pdf>
- Smith, A. B., et al. (2020). Impact of Weed Competition on Maize Growth and Yield Components under Rainfed Conditions. *Agronomy Journal*, 112(5), 3815-3826.
- Johnson, G. A., et al. (2018). Weed Control in Potato: A Review. *Agronomy Journal*, 110(2), 411-431.
- Brown, J. S., et al. (2019). Assessing the Accuracy and Reproducibility of Visual Weed Assessments in Maize. *Weed Science*, 67(5), 527-534.
- Li, H., et al. (2021). Evaluation of Machine Learning Techniques for Weed Detection in Soybean Fields Using UAV Images. *Precision Agriculture*, 22(3), 670-686.
- Craine, J. M., & Dybzinski, R. (2013). Mechanisms of plant competition for nutrients, water and light. *Functional Ecology*, 27(4), 833–840. <https://doi.org/10.1111/1365-2435.12081>
- Pusd a-Chulde, M. R., Salazar-Fierro, F. A., Sandoval-Pillajo, L., Herrera-Granda, E. P., Garc a Santill n, I. D., & de Giusti, A. (2020a). Image Analysis Based on Heterogeneous Architectures for Precision Agriculture: A Systematic Literature Review (pp. 51–70). https://doi.org/10.1007/978-3-030-33614-1_4
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer. doi:10.1007/978-3-319-24574-4_28
- Çiçek,  ., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention* (pp. 424-432). Springer.
- Kamilaris, A., Prenafeta-Bold , F. X., & Likas, A. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90. doi:10.1016/j.compag.2018.02.006
- Timar n Pereira, S. R., Hern ndez Arteaga, I., Caicedo Zambrano, S. J., Hidalgo Troya, A., & Alvarado P rez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. Universidad Cooperativa de Colombia. <https://doi.org/10.16925/9789587600490>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Garc a-Garc a, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodr guez, J. (2018). A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857.

- Microsoft. (2021). Xamarin. Retrieved
- Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(4), 75–76. <https://doi.org/10.12928/telkomnika.v14i4.3956>
- Nilsback, M. E., & Zisserman, A. (2008). Automated Flower Classification Over a Large Number of Classes. *Proceedings of the 6th Indian Conference on Computer Vision, Graphics and Image Processing*, 722–729. doi: 10.1109/ICVGIP.2008.110
- Juhász, C., Jakab, A., & Szirányi, T. (2020). Weed Detection in Precision Agriculture: A Comprehensive Review. *Sensors*, 20(10), 2852. doi: 10.3390/s20102852
- Amin, D., & Akhter, S. (2020). Deep Learning-Based Defect Detection System in Steel Sheet Surfaces. *2020 IEEE Region 10 Symposium (TENSYP)*, 5-7 June 2020, Dhaka, Bangladesh. doi: 10.1109/TENSYP50017.2020.9263783
- Smith, R. G., et al. (2020). Weed-host specificity of insects associated with invasive weeds in North America: A systematic review. *Biological Control*, 151, 104379. doi: 10.1016/j.biocontrol.2020.104379
- Pusdà-Chulde, M. R., Salazar-Fierro, F. A., Sandoval-Pillajo, L., Herrera-Granda, E. P., García-Santillán, I. D., & de Giusti, A. (2020). Competitive effect of weeds in corn crop (*Zea mays* L.). *International Journal of Plant & Soil Science*, 32(10), 54-62. doi: 10.9734/ijpss/2020/v32i1030320
- Smith, R. G., et al. (2019). Improving the efficiency of herbicide use with site-specific weed management. *Pest Management Science*, 75(11), 2951-2962. doi: 10.1002/ps.5482
- Li, C., et al. (2020). Deep learning for weed recognition in smart farming: A review. *Computers and Electronics in Agriculture*, 175, 105522. doi: 10.1016/j.compag.2020.105522
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*.
- United Nations. (2017). *The Sustainable Development Goals Report 2017*. Recuperado de <https://unstats.un.org/sdgs/files/report/2017/thesustainabledevelopmentgoalsreport2017.pdf>
- Robayo Ordoñez, A. (2022). *Construcción de un sistema de visión por computadora para la detección de maleza y líneas de cultivo utilizando imágenes provenientes de un Dron* [Tesis de pregrado, Universidad Técnica del Norte]. Recuperado de <http://repositorio.utn.edu.ec/handle/123456789/11950>
- Chicaiza Acosta, D. A. (2022). *Desarrollo de un modelo de red neuronal profunda para detección de meteoritos del sistema integrado Aallsycams en Estados Unidos* [Tesis de pregrado, Universidad Técnica del Norte]. Recuperado de

<http://repositorio.utn.edu.ec/handle/123456789/12917>

- Tene-Hurtado, D. et al. (2022). Brain Tumor Segmentation Based on 2D U-Net Using MRI Multi-modalities Brain Images. In: Narváez, F.R., Proaño, J., Morillo, P., Vallejo, D., González Montoya, D., Díaz, G.M. (eds) Smart Technologies, Systems and Applications. SmartTech-IC 2021. Communications in Computer and Information Science, vol 1532. Springer, Cham. https://doi.org/10.1007/978-3-030-99170-8_25
- Caicedo, J., Acosta, P., Pozo, R., Guilcapi, H., & Mejia-Escobar, C. (2022). Estimación de áreas de cultivo mediante Deep Learning y programación convencional. Recuperado de https://www.researchgate.net/publication/362250351_Estimacion_de_areas_de_cultivo_mediante_Deep_Learning_y_programacion_convencional
- Wang, J., Hadjikakou, M., Hewitt, R. J., & Bryan, B. A. (2022). Simulating large-scale urban land-use patterns and dynamics using the U-Net deep learning architecture. Elsevier, 97. Recuperado de <https://www.sciencedirect.com/science/article/abs/pii/S0198971522000990>
- Chicchón Apaza, M.Á., Monzón, H.M.B., Alcarria, R. (2020). Semantic Segmentation of Weeds and Crops in Multispectral Images by Using a Convolutional Neural Networks Based on U-Net. In: Botto-Tobar, M., Zambrano Vizuete, M., Torres-Carrión, P., Montes León, S., Pizarro Vásquez, G., Durakovic, B. (eds) Applied Technologies. ICAT 2019. Communications in Computer and Information Science, vol 1194. Springer, Cham. https://doi.org/10.1007/978-3-030-42520-3_38
- Kong, X., Liu, T., Chen, X., Jin, X., Li, A., & Yu, J. (2024). Efficient crop segmentation net and novel weed detection method. *European Journal of Agronomy*, 161, 127367. <https://doi.org/10.1016/j.eja.2024.127367>
- Liao, J., Chen, M., Zhang, K., Zhou, H., Zou, Y., Xiong, W., Zhang, S., Kuang, F., & Zhu, D. (2024). SC-Net: A new strip convolutional network model for rice seedling and weed segmentation in paddy field. *Computers and Electronics in Agriculture*, 220, 108862. <https://doi.org/10.1016/j.compag.2024.108862>
- Zuo, Y., & Li, W. (2024). An improved UNet lightweight network for semantic segmentation of weed images in corn fields. *Computers, Materials and Continua*, 79(3), 4413-4431. <https://doi.org/10.32604/cmc.2024.049805>
- Ma, Z., Wang, G., Yao, J., Huang, D., Tan, H., Jia, H., & Zou, Z. (2023). An improved U-Net model based on multi-scale input and attention mechanism: Application for recognition of Chinese cabbage and weed. *Sustainability (Switzerland)*, 15(7), 5764. <https://doi.org/10.3390/su15075764>
- de Silva, R., Cielniak, G., Wang, G., & Gao, J. (2024). Deep learning-based crop row detection for infield navigation of agri-robots. *Journal of Field Robotics*, 41(7), 2299-2321. <https://doi.org/10.1002/rob.22238>
- Cui J, Tan F, Bai N and Fu Y (2024) Improving U-net network for semantic segmentation of corns

and weeds during corn seedling stage in field. *Front. Plant Sci.* 15:1344958. doi: 10.3389/fpls.2024.1344958

Zhang, S., & Zhang, C. (2023). Modified U-Net for plant diseased leaf image segmentation. *Computers and Electronics in Agriculture*, 204, 107511. <https://doi.org/10.1016/j.compag.2022.107511>

Jiang, H., Chen, Q., Wang, R., Du, J., & Chen, T. (2024). SWFormer: A scale-wise hybrid CNN-Transformer network for multi-classes weed segmentation. *Journal of King Saud University - Computer and Information Sciences*, 36(7), 102144. <https://doi.org/10.1016/j.jksuci.2024.102144>

Anexos

Anexo A: Repositorio del Backend Flask

- GitHub: <https://github.com/JorgePazos-git/backend-flask>
- Google Drive: https://drive.google.com/file/d/1nr9jGcEF25wAFD6BA-PCm3zenyOxoDev/view?usp=drive_link

Anexo B: Repositorio del Frontend Xamarin

- GitHub: https://github.com/JorgePazos-git/FrontendTesisXamarin_PazosJorge.git
- Google Drive: https://drive.google.com/file/d/1I5zQ9T-pSXhwGRDip6E9hAdehKaR2uNv/view?usp=drive_link

Anexo C: Código del Preprocesamiento de Imágenes

- Google Drive: https://drive.google.com/file/d/16uo0w5b8gMI3xsYfhGB8Aa9hpPg3bA6S/view?usp=drive_link

Anexo D: Código de los Modelos Entrenados

- Google Colab: <https://colab.research.google.com/drive/1leogbTxgvL5Ukr4RDrWNqzP2-PIPWhB?usp=sharing>

Anexo E: Resultados de Cobertura por Clase

- Google Drive: https://drive.google.com/file/d/1a560VLHqa1SyHPvY-OGmXqrcwLExeWrC/view?usp=drive_link

Anexo F: Dataset Balanceado

- Google Drive: https://drive.google.com/file/d/1iqRZMHxTbqFvytqwEtXqDiN3G2S-BuCd/view?usp=drive_link

Anexo G: Dataset Desbalanceado

- Google Drive: https://drive.google.com/file/d/1EnxokIKREU_f-uWNbl_2SuCVVPWIYTL5/view?usp=drive_link